aws

Guide du développeur

Amazon MQ

Amazon MQ: Guide du développeur

Table of Contents

Qu'est-ce qu'Amazon MQ ?

Amazon MQ est un service de messagerie géré pour Apache ActiveMQ Classic et RabbitMQ qui gère la configuration, le fonctionnement et la maintenance des courtiers de messages. Vous pouvez créer un nouveau courtier Amazon MQ en utilisant les protocoles de messagerie standard du secteur ou migrer les courtiers de messages existants vers Amazon MQ sans avoir à réécrire le code de messagerie.

Un agent est un environnement d'agent de messages qui s'exécute sur Amazon MQ. Il constitue la composante de base d'Amazon MQ. Un agent de messages permet à des applications et composants logiciels de communiquer à l'aide de divers langages de programmation, systèmes d'exploitation et protocoles de messagerie formels. Vous pouvez utiliser les courtiers Amazon MQ pour la communication entre des applications et des composants cloud natifs à grande échelle.

Rubriques

Fonctionnalités d'Amazon MQ

Maintenance gérée et mises à niveau des versions

Amazon MQ effectue la maintenance et les mises à niveau de version pour un courtier de messages pendant votre période de maintenance planifiée.

Surveillez les courtiers avec CloudWatch

Amazon MQ est intégré à Amazon CloudWatch afin que vous puissiez consulter et analyser les statistiques de vos courtiers et de vos files d'attente. Vous pouvez consulter et analyser les métriques depuis la console Amazon MQ, la console, la CloudWatch ligne de commande et l'API. Les métriques sont automatiquement collectées et transmises à CloudWatch chaque minute.

Sécurité

Amazon MQ assure le chiffrement de vos messages au repos et en transit. Les connexions au broker utilisent le protocole SSL, et l'accès peut être limité à un point de terminaison privé au sein de votre Amazon VPC. De plus, vous pouvez utiliser Gestion des identités et des accès AWS(IAM) pour

contrôler les actions que vos utilisateurs et groupes IAM peuvent effectuer sur des courtiers Amazon MQ spécifiques.

Files d'attente de quorum pour RabbitMQ sur Amazon MQ

Les files d'attente de quorum sont un type de file d'attente répliqué composé d'un nœud principal (réplique principale) et de nœuds suiveurs (autres répliques). Chaque nœud se trouve dans une zone de disponibilité différente. Ainsi, si un nœud est temporairement indisponible, la livraison des messages se poursuit avec une réplique du leader nouvellement élu dans une autre zone de disponibilité. Les files d'attente de quorum sont utiles pour traiter les messages toxiques, qui apparaissent lorsqu'un message échoue et est mis en attente plusieurs fois.

Réplication de données entre régions pour ActiveMQ sur Amazon MQ

La réplication des données entre régions (CRDR) permet la réplication asynchrone des messages du courtier principal d'une région principale vers le courtier de réplication d'une AWS région de réplication. En émettant une demande de basculement à l'API Amazon MQ, l'agent de répliques actuel est promu au rôle d'agent principal et l'agent principal actuel est rétrogradé au rôle de réplique.

Comment commencer à utiliser Amazon MQ ?

Pour commencer à utiliser ActiveMQ sur Amazon MQ, consultez la documentation suivante :

- Mise en route : création et connexion à un courtier ActiveMQ
- the section called “Déploiement d'un courtier”
- Tutoriels ActiveMQ
- the section called “Bonnes pratiques Amazon MQ for ActiveMQ”

Pour commencer à utiliser RabbitMQ sur Amazon MQ, consultez la documentation suivante :

- Mise en route : création et connexion à un courtier RabbitMQ
- the section called “Déploiement d'un courtier RabbitMQ”
- the section called “Didacticiels RabbitMQ”
- the section called “Bonnes pratiques Amazon MQ for RabbitMQ”

Pour en savoir plus sur Amazon MQ REST APIs, consultez le manuel Amazon MQ REST API Reference.

Pour en savoir plus sur les AWS CLI commandes Amazon MQ, consultez Amazon MQ dans AWS CLI le manuel de référence des commandes.

Comment puis-je envoyer des commentaires à Amazon MQ ?

Nous vous invitons à nous faire part de vos commentaires sur la documentation. Vous pouvez utiliser les icônes pouces vers le haut et pouces vers le bas sur le côté droit pour envoyer des commentaires, ou vous pouvez utiliser le formulaire « Envoyer des commentaires » lié ci-dessous.

Pour contacter l'équipe Amazon MQ, utilisez le forum de discussion Amazon MQ.

Configuration d'Amazon MQ

Avant de pouvoir utiliser Amazon MQ, vous devez exécuter les tâches suivantes.

Rubriques

Étape 1 : Prérequis

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'https://portal.aws.amazon.com/billing/inscription.
2. Suivez les instructions en ligne.

   Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique ou un SMS et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à https://aws.amazon.com/et en choisissant Mon compte.

Création d’un utilisateur doté d’un accès administratif

Après vous être inscrit à un Compte AWS, sécurisez Utilisateur racine d'un compte AWS AWS IAM Identity Center, activez et créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l’aide pour vous connecter en utilisant l’utilisateur racine, consultez Connexion en tant qu’utilisateur racine dans le Guide de l’utilisateur Connexion à AWS .
2. Activez l’authentification multifactorielle (MFA) pour votre utilisateur racine.

   Pour obtenir des instructions, voir Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d’un utilisateur doté d’un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d’ AWS IAM Identity Center dans le Guide de l’utilisateur AWS IAM Identity Center .
2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu’utilisateur doté d’un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l’URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l’utilisateur IAM Identity Center.

  Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section Connexion au portail AWS d'accès dans le guide de l'Connexion à AWS utilisateur.

Attribution d’un accès à d’autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d’autorisations qui respecte la bonne pratique consistant à appliquer les autorisations de moindre privilège.

   Pour obtenir des instructions, consultez Création d’un ensemble d’autorisations dans le Guide de l’utilisateur AWS IAM Identity Center .
2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

   Pour obtenir des instructions, consultez Ajout de groupes dans le Guide de l’utilisateur AWS IAM Identity Center .

Étape 2 : créer un utilisateur et obtenir vos AWS informations d'identification

Les utilisateurs ont besoin d'un accès programmatique s'ils souhaitent interagir avec AWS l'extérieur du AWS Management Console. La manière d'accorder un accès programmatique dépend du type d'utilisateur qui y accède AWS.

Pour accorder aux utilisateurs un accès programmatique, choisissez l’une des options suivantes.

Quel utilisateur a besoin d’un accès programmatique ?	À	Méthode
IAM	(Recommandé) Utilisez les informations d'identification de la console comme informati ons d'identification temporair es pour signer les demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de l’interface que vous souhaitez utiliser. • Pour le AWS CLI, voir Connexion pour le développement AWS local dans le guide de AWS Command Line Interface l'utilisateur. • Pour AWS SDKs, voir Connexion pour le

Quel utilisateur a besoin d'un accès programmatique ?	À	Méthode
		développement AWS local dans le guide de référence AWS SDKs and Tools.
Identité de la main-d'œuvre (Utilisateurs gérés dans IAM Identity Center)	Utilisez des informations d'identification temporaires pour signer les demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de l'interface que vous souhaitez utiliser. • Pour le AWS CLI, voir Configuration du AWS CLI à utiliser AWS IAM Identity Center dans le guide de AWS Command Line Interface l'utilisateur. • Pour AWS SDKs, outils, et AWS APIs, voir Authentification IAM Identity Center dans le guide de référence AWS SDKs et Tools.
IAM	Utilisez des informations d'identification temporaires pour signer les demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de la section Utilisation d'informations d'identification temporaires avec AWS les ressources du Guide de l'utilisateur IAM.

Quel utilisateur a besoin d'un accès programmatique ?	À	Méthode
IAM	(Non recommandé) Utilisez des informations d'identification à long terme pour signer des demandes programmatiques adressées au AWS CLI AWS SDKs, ou AWS APIs.	Suivez les instructions de l'interface que vous souhaitez utiliser. • Pour le AWS CLI, voir Authentification à l'aide des informations d'identification utilisateur IAM dans le Guide de l'AWS Command Line Interface utilisateur. • Pour les outils AWS SDKs et, voir Authentifier à l'aide d'informations d'identif ication à long terme dans le guide de référence des outils AWS SDKs et. • Pour AWS APIs, voir Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Étape 3 : Se préparer à utiliser des exemples de code

Les didacticiels suivants montrent comment travailler avec les courtiers Amazon MQ en utilisant et comment vous connecter à vos courtiers Amazon MQ pour ActiveMQ et Amazon MQ pour RabbitMQ de manière programmatique. AWS Management Console Pour utiliser l'exemple de code Java ActiveMQ, vous devez installer le kit de développement Java édition Standard et apporter des modifications de code.

Vous pouvez également créer et gérer des courtiers par programmation à l'aide de l'API REST Amazon MQ et. AWS SDKs

Étapes suivantes

Maintenant que vous êtes prêt à travailler avec Amazon MQ, commencez par créer un agent. Selon le type de moteur de votre agent, vous pouvez alors connecter une application Java à votre agent Amazon MQ for ActiveMQ ou utiliser la bibliothèque client Java RabbitMQ pour connecter une application basée sur JVM à votre agent Amazon MQ for RabbitMQ.

Mise en route : création et connexion à un courtier ActiveMQ

Un agent est un environnement d'agent de messages qui s'exécute sur Amazon MQ. Il constitue la composante de base d'Amazon MQ. La description combinée de la classe d'instance de courtier (`m5`) et de la taille (`large`,`medium`) est appelée type d'instance de courtier (par exemple,`mq.m5.large`). Pour de plus amples informations, veuillez consulter Qu'est-ce qu'un courtier Amazon MQ pour ActiveMQ ?.

Création d'un courtier ActiveMQ

La tâche Amazon MQ la plus importante et la plus courante consiste à créer un agent. L'exemple suivant montre comment vous pouvez utiliser le AWS Management Console pour créer un courtier de base.

1. Connectez-vous à la console Amazon MQ.
2. Dans la page Select broker engine (Sélectionner le moteur de l'agent), choisissez Apache ActiveMQ.
3. Dans la page Select deployement and storage (Sélectionner le déploiement et le stockage), dans la section Deployement mode and storage type (Mode de déploiement et type de stockage), procédez comme suit :

   a. Cliquez sur l'onglet Deployement mode (Mode de déploiement) (par exemple, Active/standby broker (Agent actif/en veille). Pour de plus amples informations, veuillez consulter Options de déploiement pour Amazon MQ pour les courtiers ActiveMQ.

      - Un agent à instance unique est composé d'un agent dans une zone de disponibilité. L'agent communique avec votre application et avec un volume de stockage Amazon EBS ou Amazon EFS. Pour de plus amples informations, veuillez consulter Option 1 : courtiers en instance unique Amazon MQ.
      - Un agent actif/en veille pour une haute disponibilité est composé de deux agents répartis dans deux zones de disponibilité différentes, configurés dans une paire redondante. Ces agents communiquent de manière synchrone avec votre application et avec Amazon EFS. Pour de plus amples informations, veuillez consulter Option 2 : active/standby courtiers Amazon MQ pour une haute disponibilité.

   b. Cliquez sur Storage type (Type de stockage) (par exemple, EBS). Pour de plus amples informations, veuillez consulter Storage.

> **Note**
>
> Amazon EBS réplique les données dans une seule zone de disponibilité et ne prend pas en charge le mode de déploiement actif/en veille ActiveMQ.

c. Choisissez Suivant.

4. Sur la page Configure settings (Configurer les paramètres), dans la section Details (Détails), effectuez ce qui suit :

   a. Renseignez Broker name (Nom de l'agent).

> **Important**
>
> N'ajoutez pas de données d'identification personnelle (PII) ou d'autres données confidentielles ou sensibles dans les noms d'agents. Les noms des courtiers sont accessibles à d'autres AWS services, notamment CloudWatch aux journaux. Les noms d'agents ne sont pas destinées à être utilisées pour des données privées ou sensibles.

> **Note**
>
> Dans la section Paramètres supplémentaires, vous pouvez également configurer les éléments suivants :
>
> - Configurations
> - CloudWatch journaux
> - Accès privé
> - Fenêtre de maintenance du courtier

   b. Cliquez sur Broker instance type (Type d'instance de l'agent) (par exemple, mq.m5.large). Pour de plus amples informations, veuillez consulter Broker instance types.

5. Dans la section ActiveMQ Web Console access (Accès à la console web ActiveMQ), renseignez Username (Nom d'utilisateur) et Password (Mot de passe). Les restrictions suivantes s'appliquent aux noms d'utilisateur et aux mots de passe des agents :

- Votre nom d'utilisateur peut contenir uniquement des caractères alphanumériques, des tirets, des points, des traits de soulignement et des tildes (- . _ ~).
- Votre mot de passe doit comporter 12 caractères minimum, dont au moins 4 caractères uniques, et ne doit pas contenir de virgules, de deux-points ou de signes égal (,:=).

> **Important**
>
> N'ajoutez pas de données d'identification personnelle (PII) ou d'autres données confidentielles ou sensibles dans les noms d'utilisateur des agents. Les noms d'utilisateur des courtiers sont accessibles à d'autres AWS services, notamment aux CloudWatch journaux. Les noms d'utilisateur des agents ne sont pas destinées à être utilisées pour des données privées ou sensibles.

6. Choisissez Déployer.

   Alors qu'Amazon MQ crée votre agent, il affiche l'état Creation in progress (Création en cours).

   La création d'un agent prend environ 15 minutes.

   Lorsque votre agent est créé avec succès, Amazon MQ affiche l'état Running (En cours d'exécution).

7. Sélectionnez ***MyBroker***.

   Sur la ***MyBroker***page, dans la section Connect, notez l'URL de la console Web ActiveMQ de votre courtier, par exemple :

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

   Notez également que les points de terminaison du protocole de niveau filaire de votre agent. Voici un exemple de point de OpenWire terminaison :

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

Mise en route : création et connexion à un courtier RabbitMQ

Un agent est un environnement d'agent de messages qui s'exécute sur Amazon MQ. Il constitue la composante de base d'Amazon MQ. La description combinée de la classe d'instance de courtier (`m5`) et de la taille (`large`,`medium`) est appelée type d'instance de courtier (par exemple,`mq.m5.large`). Pour de plus amples informations, consultez Qu'est-ce qu'un courtier Amazon MQ pour RabbitMQ ?.

Créez un courtier RabbitMQ

La tâche Amazon MQ la plus importante et la plus courante consiste à créer un agent. L'exemple suivant montre comment vous pouvez utiliser le AWS Management Console pour créer un courtier de base.

Lorsque vous créez un courtier Amazon MQ pour RabbitMQ, suivez les meilleures pratiques de configuration du courtier pour RabbitMQ afin de maximiser les performances du courtier et d'optimiser l'efficacité du débit des messages.

1. Connectez-vous à la console Amazon MQ.
2. Dans la page Select broker engine (Sélectionner le moteur de l'agent), choisissez RabbitMQ, puis choisissez Next (Suivant).
3. Dans la page Select deployment mode (Sélectionner le mode de déploiement), choisissez le mode de déploiement, par exemple, Cluster deployment (Déploiement en cluster), puis choisissez Next (Suivant).

   - Un agent à instance unique est composé d'un agent dans une zone de disponibilité derrière d'un Network Load Balancer (NLB). L'agent communique avec votre application et avec un volume de stockage Amazon EBS. Pour de plus amples informations, veuillez consulter Option 1 : courtier à instance unique Amazon MQ pour RabbitMQ.
   - Un déploiement en cluster RabbitMQ pour une haute disponibilité est un regroupement logique de trois nœuds d'agent RabbitMQ derrière d'un Network Load Balancer, chacun partageant des utilisateurs, des files d'attente et un état distribué sur plusieurs zones de disponibilité (AZ). Pour de plus amples informations, veuillez consulter Option 2 : déploiement du cluster Amazon MQ pour RabbitMQ.
4. Sur la page Configure settings (Configurer les paramètres), dans la section Details (Détails), effectuez ce qui suit :

   a. Saisissez le nom de l'agent.

> **Important**
>
> N'ajoutez pas de données d'identification personnelle (PII) ou d'autres données confidentielles ou sensibles dans les noms d'agents. Les noms des courtiers sont accessibles à d'autres AWS services, notamment CloudWatch aux journaux. Les noms d'agents ne sont pas destinées à être utilisées pour des données privées ou sensibles.

b. Choisissez le type d'instance Broker (par exemple, mq.m7g.large). Pour de plus amples informations, veuillez consulter Broker instance types.

5. Dans la page Configure settings (Configuration des paramètres), dans la section RabbitMQ access (Accès à RabbitMQ), renseignez les champs Username (Nom d'utilisateur) et Password (Mot de passe). Les restrictions suivantes s'appliquent aux informations d'identification de connexion des agents :

   - Votre nom d'utilisateur peut contenir uniquement des caractères alphanumériques, des tirets, des points et des traits de soulignement (- . _). Cette valeur ne doit pas contenir de caractères tilde (~). Amazon MQ interdit l'utilisation de `guest` comme nom d'utilisateur.
   - Votre mot de passe doit comporter 12 caractères minimum, dont au moins 4 caractères uniques, et ne doit pas contenir de virgules, de deux-points ou de signes égal (,:=).

   > **Important**
   >
   > N'ajoutez pas de données d'identification personnelle (PII) ou d'autres données confidentielles ou sensibles dans les noms d'utilisateur des agents. Les noms d'utilisateur des courtiers sont accessibles à d'autres AWS services, notamment aux CloudWatch journaux. Les noms d'utilisateur des agents ne sont pas destinées à être utilisées pour des données privées ou sensibles.

   > **Note**
   >
   > Dans la section Paramètres supplémentaires, vous pouvez également configurer les éléments suivants :
   >
   > - Configurations

- CloudWatch journaux
- Accès privé
- Fenêtre de maintenance du courtier

6. Choisissez Suivant.
7. Dans la page Review and create (Vérifier et créer), vous pouvez vérifier vos sélections et les modifier si nécessaire.
8. Choisissez Create broker (Créer un agent).

   Alors qu'Amazon MQ crée votre agent, il affiche l'état Creation in progress (Création en cours).

   La création d'un agent prend environ 15 minutes.

   Lorsque votre agent est créé avec succès, Amazon MQ affiche l'état Running (En cours d'exécution).
9. Sélectionnez ***MyBroker***.

   Sur la ***MyBroker***page, dans la section Connect, notez l'URL de la console Web RabbitMQ de votre courtier, par exemple :

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

   Notez également le point de terminaison secure-AMQP de votre agent. Voici un exemple de `amqps`point de terminaison exposant un port d'écouteur `5671`.

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

Gestion d'un agent Amazon MQ

Après avoir créé un courtier, vous pouvez gérer et maintenir les différents composants de votre courtier Amazon MQ.

Rubriques

Connexion à Amazon MQ

Vous pouvez vous connecter à Amazon MQ depuis d'autres AWS services à l'aide de points de terminaison de service et de points de terminaison de courtier.

Points de terminaison de service

Les méthodes de connexion suivantes sont utilisées pour l'API du service Amazon MQ :

Domaines	Méthode de connexion
`mq.region.amazonaws.com`	IPv4
`mq.region.api.aws`	Dual-Stack (IPv4 et IPv6)
`mq-fips.region.amazonaws.com`	FIPS avec uniquement IPv4

Domaines	Méthode de connexion
`mq-fips.region.api.aws`	FIPS avec double pile

Points de terminaison du courtier

Les méthodes de connexion suivantes sont utilisées par les courtiers Amazon MQ :

Domaines	Méthode de connexion
`brokerId.mq.region.amazonaws.com`	IPv4
`brokerId.mq.region.on.aws`	Dual-Stack (IPv4 et IPv6) **Note** Les courtiers Amazon MQ pour ActiveMQ ne prennent pas en charge le double stack.

Connectez-vous à Amazon MQ à l'aide de points de terminaison à double pile (IPv4 et) IPv6

Les terminaux à double pile prennent en charge à la fois le trafic IPv4 et IPv6 le trafic. Lorsque vous envoyez une demande à un point de terminaison à double pile, l'URL du point de terminaison est une adresse IPv4 ou une IPv6 adresse. Pour plus d'informations sur les points de terminaison à double pile et FIPS, consultez le guide de référence du SDK.

Amazon MQ prend en charge les points de terminaison régionaux à double pile, ce qui signifie que vous devez spécifier la AWS région dans le nom du point de terminaison. Les noms de point de terminaison à double pile utilisent la convention de dénomination suivante :`mq.region.api.aws`. Par exemple, le nom du point de terminaison à double pile de la région `eu-west-1` est `mq.eu-west-1.api.aws`.

Pour obtenir la liste complète des points de terminaison Amazon MQ, consultez la AWS référence générale.

Connectez-vous à Amazon MQ à l'aide de AWS PrivateLink

AWS PrivateLinkpoints de terminaison pour l'API Amazon MQ prenant en charge IPv4 IPv6 et fournissant une connectivité privée entre les clouds privés virtuels VPCs () et l'API Amazon MQ sans exposer votre trafic à l'Internet public.

> **Note**
>
> Support uniquement PrivateLink disponible pour le point de terminaison de l'API Amazon MQ, et non pour le point de terminaison du courtier. Pour plus d'informations sur la connexion privée à un point de terminaison de courtier, consultezConfiguring a private Amazon MQ broker.

Pour accéder à l'API Amazon MQ à l'aide de l'API PrivateLink, vous devez d'abord créer un point de terminaison VPC d'interface dans le VPC spécifique à partir duquel vous souhaitez vous connecter. Lorsque vous créez le point de terminaison VPC, utilisez le nom du service `com.amazonaws.region.mq` ou `com.amazonaws.region.mq-fips` pour les points de terminaison FIPS.

Lorsque vous appelez Amazon MQ à l'aide de la AWS CLI ou du SDK, vous devez spécifier l'URL du point de terminaison pour utiliser le nom de domaine à double pile : ou. `mq.region.api.aws` `mq-fips.region.api.aws` PrivateLink pour Amazon MQ ne prend pas en charge le nom de domaine par défaut se terminant par. `amazonaws.com` Pour plus d'informations, consultez la section Points de terminaison à double pile et FIPS dans le guide de référence du SDK.

L'exemple de CLI suivant montre comment appeler le `describe-broker-engine-type` dans la région Asie-Pacifique (Sydney) via un point de terminaison Amazon MQ VPC.

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

Pour d'autres méthodes de configuration du point de terminaison dans la CLI, voir Utilisation des points de terminaison dans la AWS CLI

Vous pouvez également déterminer l'accès des utilisateurs aux points de terminaison VPC à l'aide des politiques de point de terminaison VPC. Pour plus d'informations, consultez Contrôler l'accès aux points de terminaison VPC à l'aide de politiques de point de terminaison.

Authentification et autorisation pour les courtiers Amazon MQ

Amazon MQ propose plusieurs méthodes d'authentification et d'autorisation pour sécuriser votre infrastructure de messagerie conformément aux exigences de votre entreprise.

Authentification et autorisation pour Amazon MQ pour RabbitMQ

Amazon MQ pour RabbitMQ prend en charge les méthodes d'authentification et d'autorisation suivantes :

Authentification et autorisation simples

Dans cette méthode, les utilisateurs du courtier sont stockés en interne dans le courtier RabbitMQ et gérés via la console Web ou l'API de gestion. Les autorisations pour les hôtes virtuels, les échanges, les files d'attente et les sujets sont configurées directement dans RabbitMQ. Il s'agit de la méthode par défaut. Pour plus d'informations, consultez Authentification et autorisation simples.

OAuth Authentification et autorisation 2.0

Dans cette méthode, les utilisateurs du broker et leurs autorisations sont gérés par un fournisseur d'identité OAuth 2.0 externe (IdP). L'authentification des utilisateurs et les autorisations de ressources pour les hôtes virtuels, les échanges, les files d'attente et les sujets sont centralisées via le système de périmètre du fournisseur OAuth 2.0. Cela simplifie la gestion des utilisateurs et permet l'intégration aux systèmes d'identité existants. Pour plus d'informations, voir Authentification et autorisation OAuth 2.0.

Authentification et autorisation IAM

Dans cette méthode, les utilisateurs du broker s'authentifient à l'aide des informations d'identification AWS IAM via la fédération sortante IAM. Les informations d'identification IAM sont utilisées pour obtenir des jetons JWT auprès du AWS Security Token Service (STS), et ces jetons JWT servent de jetons OAuth 2.0 pour l'authentification. Cette méthode s'appuie sur le support OAuth 2.0 existant dans Amazon MQ pour RabbitMQ, qui agit en tant que fournisseur d' AWS identité 2.0. OAuth L'authentification des utilisateurs est gérée par AWS IAM, tandis que les autorisations de ressources pour les hôtes virtuels, les échanges, les files d'attente et les sujets sont gérées via des politiques IAM et des alias de portée configurés dans RabbitMQ. Pour plus d'informations, consultez Authentification et autorisation IAM.

Authentification et autorisation LDAP

Dans cette méthode, les utilisateurs du broker et leurs autorisations sont gérés par un service d'annuaire LDAP externe. L'authentification des utilisateurs et les autorisations de ressources sont centralisées via le serveur LDAP, ce qui permet aux utilisateurs d'accéder à RabbitMQ en utilisant leurs informations d'identification de service d'annuaire existantes. Pour plus d'informations, consultez Authentification et autorisation LDAP.

Authentification et autorisation HTTP

Dans cette méthode, les utilisateurs du broker et leurs autorisations sont gérés par un serveur HTTP externe. L'authentification des utilisateurs et les autorisations de ressources sont centralisées via le serveur HTTP, ce qui permet aux utilisateurs d'accéder à RabbitMQ en utilisant leur propre fournisseur d'authentification et d'autorisation. Pour plus d'informations sur cette méthode, consultez Authentification et autorisation HTTP.

Authentification par certificat SSL

Amazon MQ prend en charge le protocole TLS mutuel (MTL) pour les courtiers RabbitMQ. Le plugin d'authentification SSL utilise des certificats clients issus de connexions mTLS pour authentifier les utilisateurs. Dans cette méthode, les utilisateurs du broker sont authentifiés à l'aide de certificats clients X.509 au lieu de leur nom d'utilisateur et de leur mot de passe. Le certificat du client est validé auprès d'une autorité de certification (CA) fiable, et le nom d'utilisateur est extrait d'un champ du certificat, tel que le nom commun (CN) ou le nom alternatif du sujet (SAN). Cette méthode fournit une authentification forte sans transmettre d'informations d'identification sur le réseau. Pour plus d'informations, consultez la section Authentification par certificat SSL.

> **Note**
>
> RabbitMQ prend en charge plusieurs méthodes d'authentification et d'autorisation à utiliser simultanément. Par exemple, vous pouvez activer à la fois l'authentification OAuth 2.0 et l'authentification simple (interne). Pour plus d'informations, consultez la section du didacticiel OAuth 2.0 sur l'activation à la fois de l'authentification OAuth 2.0 et de l'authentification simple (interne) et la documentation sur le contrôle d'accès RabbitMQ.
> Amazon MQ recommande de créer un utilisateur interne lors du test des configurations d'authentification. Cela permet de valider la configuration des accès à l'aide de l'API de gestion RabbitMQ. Pour plus d'informations, consultez la section Validation des accès.

Authentification et autorisation pour Amazon MQ pour ActiveMQ

Amazon MQ pour ActiveMQ prend en charge les méthodes d'authentification et d'autorisation suivantes :

Authentification et autorisation simples

Dans cette méthode, les utilisateurs du broker sont créés et gérés via la console ou l'API Amazon MQ. Les utilisateurs peuvent être configurés avec des autorisations spécifiques pour accéder aux files d'attente, aux rubriques et à la console Web ActiveMQ. Pour plus d'informations sur cette méthode, consultez la section Création d'un utilisateur de courtier ActiveMQ.

Authentification et autorisation LDAP

Dans cette méthode, les utilisateurs du broker s'authentifient à l'aide des informations d'identification stockées sur votre serveur LDAP. Vous pouvez ajouter, supprimer et modifier des utilisateurs et attribuer des autorisations aux sujets et aux files d'attente via le serveur LDAP, en fournissant une authentification et une autorisation centralisées. Pour plus d'informations sur cette méthode, consultez la section Intégration des courtiers ActiveMQ à LDAP.

Mise à niveau d'une version du moteur d'agent Amazon MQ

Amazon MQ propose régulièrement de nouvelles versions de moteurs de courtage pour tous les types de moteurs de courtage pris en charge. Les nouvelles versions du moteur incluent des correctifs de sécurité, des corrections de bogues et d'autres améliorations du moteur de courtage.

Amazon MQ organise les numéros de version en fonction des spécifications de version sémantiques sous la forme. `X.Y.Z` Dans les implémentations d'Amazon MQ, `X` indique la version majeure, `Y` représente la version mineure et `Z` indique le numéro de version du correctif. Amazon MQ prend en charge deux types de mises à niveau :

- Mise à niveau de version majeure – Survient lorsque les numéros de version majeure du moteur changent. Par exemple, la mise à niveau de RabbitMQ version 3.13 vers la version 4.2 est considérée comme une mise à niveau de version majeure.
- Mise à niveau de version mineure – Survient lorsque seul le numéro de version du moteur mineur change. Par exemple, mise à niveau depuis la version 3. 11 à la version 3. La version 12 est considérée comme une mise à niveau mineure.

Vous pouvez à tout moment mettre à niveau manuellement votre courtier vers la prochaine version majeure ou mineure prise en charge. Amazon MQ gère la mise à niveau vers la dernière version de correctif prise en charge pour tous les courtiers pendant la période de maintenance planifiée. Les mises à niveau de version manuelles et automatiques ont lieu pendant la période de maintenance planifiée ou après le redémarrage de votre courtier. Amazon MQ met à niveau votre courtier vers la version mineure suivante lorsque le support de la version mineure actuelle atteint la fin du support.

Mise à niveau manuelle de la version du moteur

Vous pouvez mettre à niveau la version du moteur d'un broker à l'aide de l' AWS Management Console API, de AWS CLI, ou de l'API Amazon MQ.

AWS Management Console

Pour mettre à niveau la version du moteur d'un broker à l'aide du AWS Management Console

1. Sur la page de détails de l'agent, choisissez Edit (Modifier).
2. Sous Specifications (Spécifications), pour Broker engine version (Version du moteur de l'agent) choisissez le numéro de la nouvelle version dans la liste déroulante.
3. Faites défiler l'écran jusqu'au bas de la page et choisissez Schedule modifications (Planifier les modifications).
4. Dans la page Schedule broker modifications (Planifier les modifications de l'agent), pour When to apply modifications (Quand appliquer les modifications), choisissez l'une des options suivantes.

   - Choisissez After the next reboot (Après le prochain redémarrage), si vous souhaitez qu'Amazon MQ effectue la mise à niveau de version lors de la prochaine fenêtre de maintenance planifiée.
   - Choisissez Immediately (Immédiatement), si vous souhaitez redémarrer l'agent et mettre à niveau la version du moteur immédiatement.

   > **Important**
   >
   > Les courtiers à instance unique sont hors ligne lors du redémarrage. Pour les courtiers en clusters, un seul nœud est en panne à la fois lorsque le courtier redémarre.

5. Choisissez Apply (Appliquer)pour terminer l'application des modifications.

AWS CLI

Pour mettre à niveau la version du moteur d'un broker à l'aide du AWS CLI

1. Utilisez la commande CLI update-broker et spécifiez les paramètres suivants, comme illustré dans l'exemple.

   - `--broker-id` – ID unique généré par Amazon MQ pour l'agent. Vous pouvez analyser l'ID de votre ARN d'agent. Par exemple, avec l'ARN suivant, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, l'ID de l'agent serait `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
   - `--engine-version` – Numéro de version du moteur de l'agent vers lequel effectuer la mise à niveau.

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

2. (Facultatif) Utilisez la commande CLI reboot-broker pour redémarrer votre broker si vous souhaitez mettre immédiatement à niveau la version du moteur.

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

   Si vous ne souhaitez pas redémarrer votre agent et appliquer les modifications immédiatement, Amazon MQ mettra à niveau l'agent au cours de la prochaine fenêtre de maintenance planifiée.

   > **Important**
   >
   > Les courtiers à instance unique sont hors ligne lors du redémarrage. Pour les courtiers en clusters, un seul nœud est en panne à la fois lorsque le courtier redémarre.

API Amazon MQ

Pour mettre à niveau la version du moteur d'un agent à l'aide de l'API Amazon MQ

1. Utilisez l'opération d'API UpdateBroker. Précisez `broker-id` comme un paramètre de chemin. Les exemples suivants supposent qu'un agent est dans la région `us-west-2`. Pour de plus amples informations sur les points de terminaison Amazon MQ, consultez Quotas et points de terminaison Amazon MQ dans la Références générales 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
```

Utilisez `engineVersion` dans la charge utile de la demande pour spécifier le numéro de la version vers laquelle l'agent doit effectuer la mise à niveau.

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

2. (Facultatif) Utilisez l'opération RebootBrokerAPI pour redémarrer votre broker si vous souhaitez mettre à niveau la version du moteur immédiatement. `broker-id`est spécifié en tant que paramètre de chemin.

```
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
```

Si vous ne souhaitez pas redémarrer votre agent et appliquer les modifications immédiatement, Amazon MQ mettra à niveau l'agent au cours de la prochaine fenêtre de maintenance planifiée.

> **Important**
>
> Les courtiers à instance unique sont hors ligne lors du redémarrage. Pour les courtiers en clusters, un seul nœud est en panne à la fois lorsque le courtier redémarre.

Mise à niveau d'un type d'instance de courtier Amazon MQ

> **Important**
>
> `mq.m7g.x`les instances ne sont disponibles que pour Amazon MQ pour les courtiers RabbitMQ. Les courtiers Amazon MQ pour ActiveMQ utilisent uniquement des instances. `mq.m5.x`

La description combinée de la classe d'instance de courtier (`m7g`) et de la taille (`large`) est appelée type d'instance de courtier (par exemple,`mq.m7g.large`). Lorsque vous choisissez un type d'instance, il est important de prendre en compte les facteurs qui affecteront les performances du courtier :

- le nombre de clients et de files d'attente
- le volume de messages envoyés
- messages conservés en mémoire
- messages redondants

Les types d'instances de broker plus petits (`mq.m7g.medium`) sont recommandés uniquement pour tester les performances des applications. Nous recommandons des types d'instances de broker plus grands (`mq.m7g.large` et supérieurs) pour les niveaux de production des clients et des files d'attente, le débit élevé, les messages en mémoire et les messages redondants.

Nous vous recommandons de passer à un type d'instance plus important (c'est-à-dire de `micro` vers`large`) si vous rencontrez des problèmes de performances ou si vous passez d'un environnement de test à un environnement de production. Pour mettre à niveau votre type d'instance, vous pouvez utiliser l'API AWS Management Console AWS CLI, la ou l'API Amazon MQ.

AWS Management Console

Pour effectuer une mise à niveau vers un type d'instance plus important à l'aide du AWS Management Console, procédez comme suit :

1. Connectez-vous à la console Amazon MQ.
2. Dans le panneau de navigation de gauche, choisissezBrokers (Agents), puis choisissez l’agent que vous souhaitez mettre à niveau dans la liste.

3. Sur la page Details (Informations), choisissez Edit (Modifier).
4. Sous Spécifications, pour le type d'instance Broker, choisissez le nouveau type d'instance dans la liste déroulante.
5. Au bas de la page, sélectionnez Planifier les modifications.
6. Dans la page Schedule broker modifications (Planifier les modifications de l'agent), pour When to apply modifications (Quand appliquer les modifications), choisissez l'une des options suivantes.

   - Choisissez Après le prochain redémarrage, si vous souhaitez qu'Amazon MQ termine la mise à niveau lors de la prochaine fenêtre de maintenance planifiée.
   - Choisissez Immédiatement si vous souhaitez redémarrer le broker et mettre à niveau le type d'instance immédiatement.

   > **Important**
   >
   > Les courtiers à instance unique sont hors ligne lors du redémarrage. Pour les courtiers en clusters, un seul nœud est en panne à la fois lorsque le courtier redémarre.

7. Choisissez Apply (Appliquer)pour terminer l'application des modifications.

AWS CLI

Pour mettre à niveau le type d'instance d'un courtier à l'aide du AWS CLI

1. Utilisez la commande modify-broker CLI et spécifiez les paramètres suivants, comme indiqué dans l'exemple.

   - `--broker-id` – ID unique généré par Amazon MQ pour l'agent.
   - `--host-instance-type` – Numéro de version du moteur de l'agent vers lequel effectuer la mise à niveau.

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

2. (Facultatif) Utilisez la commande CLI reboot-broker pour redémarrer votre courtier si vous souhaitez mettre à niveau le type d'instance immédiatement.

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

Si vous ne souhaitez pas redémarrer votre agent et appliquer les modifications immédiatement, Amazon MQ mettra à niveau l'agent au cours de la prochaine fenêtre de maintenance planifiée.

> **Important**
>
> Les courtiers à instance unique sont hors ligne lors du redémarrage. Pour les courtiers en clusters, un seul nœud est en panne à la fois lorsque le courtier redémarre.

API Amazon MQ

Pour mettre à niveau le type d'instance d'un broker à l'aide de l'API Amazon MQ

1. Utilisez l'opération d'API UpdateBroker. Précisez `broker-id` comme un paramètre de chemin. Les exemples suivants supposent qu'un agent est dans la région `us-west-2`. Pour plus d'informations sur les points de terminaison Amazon MQ disponibles, consultez la section Points de terminaison et quotas Amazon MQ dans le. Références générales 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
```

   `host-instance-type`À utiliser dans la charge utile de la demande pour spécifier le type d'instance vers lequel le broker doit effectuer la mise à niveau.

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

2. (Facultatif) Utilisez l'opération RebootBrokerAPI pour redémarrer votre broker, si vous souhaitez mettre à niveau la version du moteur immédiatement. `broker-id`est spécifié en tant que paramètre de chemin.

```
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
```

Si vous ne souhaitez pas redémarrer votre agent et appliquer les modifications immédiatement, Amazon MQ mettra à niveau l'agent au cours de la prochaine fenêtre de maintenance planifiée.

> **Important**
>
> Les courtiers à instance unique sont hors ligne lors du redémarrage. Pour les courtiers en clusters, un seul nœud est en panne à la fois lorsque le courtier redémarre.

Amazon MQ pour les types de stockage ActiveMQ

Amazon MQ for ActiveMQ prend en charge Amazon Elastic File System (EFS) et Amazon Elastic Block Store (EBS). Par défaut, les agents ActiveMQ utilisent Amazon EFS pour le stockage d'agents. Pour tirer parti d'une grande durabilité et d'une réplication sur plusieurs zones de disponibilité, utilisez Amazon EFS. Pour profiter d'une faible latence et d'un débit élevé, utilisez Amazon EBS.

> **Important**
>
> - Vous pouvez utiliser Amazon EBS uniquement avec la gamme de type d'instance d'agent `mq.m5`.
> - Bien que vous puissiez modifier le type d'instance de l'agent, vous ne pouvez pas modifier le type de stockage de l'agent après avoir créé l'agent.
> - Amazon EBS réplique les données dans une seule zone de disponibilité et ne prend pas en charge le mode de déploiement actif/en veille ActiveMQ.

Différences entre les types de stockage

Le tableau suivant présente brièvement les différences entre les types de stockage en mémoire, Amazon EFS et Amazon EBS pour les agents ActiveMQ.

Storage Type	Persistance	Exemple de cas d'utilisation	Nombre maximal approximatif de messages mis en file d'attente par producteur et par seconde (message de 1 Ko)	Réplication
En mémoire	Non persistant	• Cotations boursières • Mise à jour de données de localisation • Données fréquemment modifiées	5 000	Aucune
Amazon EBS	Persistante	• Volumes importants de texte • Traitement de commandes	500	Copies multiples au sein d'une même zone de disponibilité (AZ)
Amazon EFS	Persistante	Transactions financières	80	Plusieurs copies sur plusieurs AZs

Le stockage de messages en mémoire offre la latence la plus faible et le débit le plus élevé. Toutefois, les messages sont perdus en cas de remplacement de l'instance ou du redémarrage de l'agent.

Amazon EFS est conçu pour être extrêmement durable, répliqué sur plusieurs AZs sites afin d'éviter la perte de données résultant de la défaillance d'un composant ou d'un problème affectant la disponibilité d'un AZ. Amazon EBS est optimisé pour le débit et répliqué sur plusieurs serveurs dans une même zone de disponibilité.

Configuration d'un courtier Amazon MQ privé

Un courtier privé n'est pas accessible au public et n'est pas accessible depuis l'extérieur de votre VPC. Avant de configurer un courtier privé, consultez les informations suivantes sur VPCs les sous-réseaux et les groupes de sécurité :

- VPCs
  - Le ou les sous-réseaux et groupes de sécurité d'un broker doivent se trouver dans le même VPC.
  - Lorsque vous utilisez un courtier privé, vous pouvez voir des adresses IP que vous n'avez pas configurées avec votre VPC. Il s'agit d'adresses IP provenant de l'infrastructure Amazon MQ, qui ne nécessitent aucune action.
- Sous-réseaux
  - Si des sous-réseaux se trouvent au sein d'un VPC partagé, le VPC doit appartenir au même compte qui a créé le courtier.
  - Si aucun sous-réseau n'est fourni, les sous-réseaux par défaut du VPC par défaut seront utilisés.
  - Une fois le broker créé, les sous-réseaux utilisés ne peuvent pas être modifiés.
  - Pour les clusters et les active/standby courtiers, les sous-réseaux doivent se trouver dans des zones de disponibilité différentes.
  - Pour les courtiers à instance unique, vous pouvez spécifier le sous-réseau à utiliser et le courtier sera créé dans la même zone de disponibilité.
- Groupes de sécurité
  - Si aucun groupe de sécurité n'est fourni, les groupes de sécurité par défaut du VPC par défaut seront utilisés.
  - Les instances uniques, les clusters et active/standby les courtiers nécessitent au moins un groupe de sécurité (par exemple, le groupe de sécurité par défaut).

    > **Note**
    >
    > Les courtiers publics de RabbitMQ n'utilisent pas de sous-réseaux ou de groupes de sécurité.

  - Une fois le broker créé, le groupe de sécurité utilisé ne peut pas être modifié. Les groupes de sécurité eux-mêmes peuvent toujours être modifiés.

Configuration d'un courtier privé dans AWS Management Console

Pour configurer un courtier privé, commencez à créer un nouveau courtier dans le AWS Management Console. Ensuite, dans la section Paramètres réseau, pour configurer la connectivité de votre courtier, procédez comme suit :

1. Choisissez l'accès privé pour votre courtier. Pour vous connecter à un courtier privé, vous pouvez utiliser IPv4 IPv6, ou Dual-Stack (IPv4 et IPv6). Pour de plus amples informations, veuillez consulter Connecting to Amazon MQ.
2. Ensuite, choisissez Utiliser le VPC, le ou les sous-réseaux et les groupes de sécurité par défaut, ou sélectionnez Sélectionner un VPC, un ou plusieurs sous-réseaux et groupes de sécurité existants. Si vous ne souhaitez pas utiliser le VPC, sous-réseau ou groupe de sécurité par défaut ou existant, vous devez en créer un nouveau pour vous connecter au courtier privé.

> **Note**
>
> Pour l'accès à un courtier privé, la méthode de connexion sera la même que le type d'adresse IP sélectionné pour le sous-réseau. Une fois le broker créé, le point de terminaison VPC ne peut pas être modifié et contiendra toujours le type IP des sous-réseaux sélectionnés. Si vous souhaitez utiliser un nouveau type d'adresse IP, vous devez créer un nouveau courtier.

> **Note**
>
> Amazon MQ pour ActiveMQ n'utilise pas de points de terminaison VPC. Lorsque vous créez un broker ActiveMQ pour la première fois, Amazon MQ fournit une interface ELASTIC (ENI) dans le VPC. Les groupes de sécurité sont placés dans l'ENI et peuvent être utilisés à la fois par les courtiers publics et privés.

Accès à la console Web du courtier Amazon MQ sans accès public

Lorsque vous désactivez l'accessibilité publique pour votre courtier, l'identifiant de AWS compte qui a créé le courtier peut accéder au courtier privé. Si vous désactivez l'accessibilité publique pour votre courtier, vous devez effectuer les étapes suivantes pour accéder à la console Web du courtier.

1. Créez une instance EC2 Linux dans `public-vpc` (avec une adresse IP publique, si nécessaire).

2. Pour vérifier que votre VPC est configuré correctement, établissez une connexion `ssh` à l'instance EC2 et utilisez la commande `curl` avec l'URI de votre agent.
3. Depuis votre ordinateur, créez un tunnel `ssh` vers l'instance EC2 en utilisant le chemin d'accès à votre fichier de clé privée et l'adresse IP de votre instance EC2 public. Par exemple :

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

   Un serveur proxy de réacheminement est démarré sur votre ordinateur.
4. Installez un client proxy, par exemple FoxyProxysur votre machine.
5. Configurez votre client proxy en utilisant les paramètres suivants :

   - Pour le type de proxy, spécifiez `SOCKS5`.
   - Pour l'adresse IP, le nom DNS et le nom du serveur, spécifiez `localhost`.
   - Pour un port, spécifiez `8080`.
   - Supprimez les modèles d'URL existants.
   - Pour le modèle d'URL, spécifiez `*.mq.*.amazonaws.com*`.
   - Pour le type de connexion, spécifiez `HTTP(S)`.

   Lorsque vous activez votre client proxy, vous pouvez accéder à la console web sur votre ordinateur.

> **Important**
>
> Si vous utilisez un courtier privé, vous pouvez voir des adresses IP que vous n'avez pas configurées avec votre VPC. Il s'agit d'adresses IP provenant de l'infrastructure RabbitMQ sur Amazon MQ, et elles ne nécessitent aucune action.

Planification de la fenêtre de maintenance pour un courtier Amazon MQ

Amazon MQ effectue régulièrement la maintenance du matériel, du système d'exploitation ou du logiciel moteur d'un courtier de messages pendant la période de maintenance. Par exemple, si vous avez modifié le type d'instance de courtier, Amazon MQ appliquera vos modifications lors

de la prochaine fenêtre de maintenance planifiée. La durée de la maintenance peut durer jusqu'à deux heures en fonction des opérations planifiées pour votre courtier de messages. Vous pouvez minimiser les temps d'arrêt pendant une période de maintenance en sélectionnant un mode de déploiement de courtier offrant une haute disponibilité sur plusieurs zones de disponibilité (AZ).

Amazon MQ pour ActiveMQ propose des déploiements actifs/en veille pour une haute disponibilité. En active/standby mode, Amazon MQ effectue les opérations de maintenance une instance à la fois, et au moins une instance reste disponible. De plus, vous pouvez configurer un réseau de courtiers avec des fenêtres de maintenance variées au cours de la semaine. Amazon MQ pour RabbitMQ fournit les déploiements de clusters pour une haute disponibilité. Dans les déploiements de clusters, Amazon MQ effectue les opérations de maintenance un nœud à la fois en conservant au moins deux nœuds actifs à tout moment.

Lorsque vous créez votre courtier pour la première fois, vous pouvez planifier la période de maintenance une fois par semaine à une heure précise. Vous ne pouvez ajuster la fenêtre de maintenance d’un agent que quatre fois maximum avant la prochaine fenêtre de maintenance planifiée. Une fois la période de maintenance d'un courtier terminée, Amazon MQ réinitialise la limite et vous pouvez à nouveau ajuster le calendrier avant le début de la fenêtre de maintenance suivante. La disponibilité des courtiers n'est pas affectée lors de l'ajustement de la fenêtre de maintenance des courtiers.

Pour ajuster la fenêtre de maintenance des courtiers, vous pouvez utiliser l' AWS Management Console API AWS CLI, la ou l'API Amazon MQ.

Planifiez la fenêtre de maintenance du courtier à l'aide du AWS Management Console

Pour ajuster la fenêtre de maintenance du courtier à l'aide du AWS Management Console

1. Connectez-vous à la console Amazon MQ.
2. Dans le panneau de navigation de gauche, choisissezBrokers (Agents), puis choisissez l’agent que vous souhaitez mettre à niveau dans la liste.
3. Sur la page Details (Informations), choisissez Edit (Modifier).
4. Sous Maintenance, procédez comme suit :

    a. Pour Start day (Jour de début), choisissez un jour de la semaine, par exemple Sunday (Dimanche), dans la liste déroulante.

    b. Pour Start time (Heure de début), choisissez l’heure et la minute que vous souhaitez définir pour la prochaine fenêtre de maintenance de l’agent, par exemple 12:00.

> **Note**
>
> Les options Start time (Heure de début) sont configurées selon le fuseau horaire UTC+0.

5. Ensuite, sélectionnez Modifications du calendrier. Choisissez ensuite Après le prochain redémarrage ou Immédiatement. Si vous choisissez Après le prochain redémarrage, la fenêtre de maintenance sera immédiatement mise à jour sans redémarrer le broker. Si vous sélectionnez Immédiatement, le courtier sera immédiatement redémarré.
6. Sur la page d'informations de l'agent, sous Maintenance window (Fenêtre de maintenance), vérifiez que votre nouvelle préférence de planification s'affiche.

Planifiez la fenêtre de maintenance du courtier à l'aide du AWS CLI

Pour ajuster la fenêtre de maintenance du courtier à l'aide du AWS CLI

1. Utilisez la commande CLI update-broker et spécifiez les paramètres suivants, comme illustré dans l'exemple.

   - `--broker-id` – ID unique généré par Amazon MQ pour l'agent. Vous pouvez analyser l'ID de votre ARN d'agent. Par exemple, avec l'ARN suivant, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, l'ID de l'agent serait `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
   - `--maintenance-window-start-time` – Les paramètres qui déterminent l'heure de début de la fenêtre de maintenance hebdomadaire fournie dans la structure suivante.
     - `DayOfWeek` – Le jour de la semaine, dans la syntaxe suivante : `MONDAY| TUESDAY | WEDNESDAY | THURSDAY | FRIDAY | SATURDAY | SUNDAY`
     - `TimeOfDay` – L'heure au format 24 heures.
     - `TimeZone` – (Facultatif) Le fuseau horaire, soit au format Pays/Ville, soit en indiquant le décalage par rapport à l'heure UTC. Définie sur UTC par défaut.

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

2. (Facultatif) Utilisez la commande de CLI describe-agent pour vérifier que la fenêtre de maintenance est correctement mise à jour.

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

Planifiez la fenêtre de maintenance du courtier à l'aide de l'API Amazon MQ

Pour ajuster la fenêtre de maintenance de l'agent à l'aide de l'API Amazon MQ

1. Utilisez l'opération d'API UpdateBroker. Précisez `broker-id` comme un paramètre de chemin. Les exemples suivants supposent qu'un agent est dans la région `us-west-2`. Pour plus d'informations sur les points de terminaison Amazon MQ disponibles, consultez la section Points de terminaison et quotas Amazon MQ dans le. Références générales 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
```

   Utilisez le paramètre `maintenanceWindowStartTime` et le type de ressource `WeeklyStartTime` dans la charge utile de la demande.

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

2. (Facultatif) Utilisez l'opération DescribeBrokerAPI pour vérifier que la fenêtre de maintenance a été correctement mise à jour. `broker-id`est spécifié en tant que paramètre de chemin.

```
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
```

Redémarrage d'un agent Amazon MQ

Pour appliquer une nouvelle configuration à un agent, vous pouvez redémarrer l'agent.

> **Note**
>
> Si votre agent ActiveMQ cesse de répondre, vous pouvez le redémarrer pour le sortir d'un état défectueux.

L'exemple suivant montre comment redémarrer un agent Amazon MQ à l'aide de la AWS Management Console.

Pour redémarrer un agent Amazon MQ

1. Connectez-vous à la console Amazon MQ.
2. Dans la liste des courtiers, choisissez le nom de votre courtier (par exemple, MyBroker).
3. Sur la ***MyBroker***page, choisissez Actions, Redémarrer le courtier.

   > **Important**
   >
   > Les agents à instance unique seront hors ligne lors de leur redémarrage. Les agents en cluster seront disponibles, mais chaque nœud est redémarré l'un après l'autre.

4. Dans la boîte de dialogue Reboot broker, choisissez Reboot.

   Rebooting a broker takes about 5 minutes. Si le redémarrage inclut des modifications de la taille de l'instance ou s'il est effectué sur un agent dont la longueur de file d'attente est élevée, le processus de redémarrage peut prendre plus de temps.

Suppression d'un agent Amazon MQ

Si vous n'utilisez pas de courtier Amazon MQ (et ne prévoyez pas de l'utiliser dans un futur proche), il est recommandé de le supprimer d'Amazon MQ afin de réduire vos coûts. AWS

L'exemple suivant montre comment supprimer un agent à l'aide de la AWS Management Console.

Suppression d'un agent Amazon MQ

1. Connectez-vous à la console Amazon MQ.
2. Dans la liste des courtiers, sélectionnez votre courtier (par exemple MyBroker), puis choisissez Supprimer.
3. Dans le champ Supprimer ***`MyBroker`*** ? dans une boîte de dialogue, tapez `delete` puis choisissez Supprimer.

   La suppression d'un agent prend environ 5 minutes.

Statuts des courtiers Amazon MQ

La condition actuelle d'un agent est indiquée par un statut. Le tableau suivant répertorie les états d'un agent Amazon MQ.

Console	API	Description
Échec de la création	`CREATION_FAILED`	Impossible de créer l'agent.
Création en cours	`CREATION_IN_PROGRESS`	L'agent est en cours de création.
Suppression en cours	`DELETION_IN_PROGRESS`	L'agent est en cours de suppression.
Redémarrage en cours	`REBOOT_IN_PROGRESS`	L'agent est en cours de redémarrage.
En cours d’exécution	`RUNNING`	L'agent est opérationnel.
Action critique requise	`CRITICAL_ACTION_REQUIRED`	L’agent est en cours d'exécution, mais il est dans un état dégradé et nécessite une action immédiate. Vous trouverez des instructions pour résoudre le problème en utilisant le code d'action

Console	API	Description
		requise dans la liste de Résolution des problèmes.

Ajouter des balises aux ressources Amazon MQ

Pour organiser et identifier vos ressources Amazon MQ pour la répartition des coûts, vous pouvez ajouter des balises de métadonnées qui identifient le but d'un agent ou d'une configuration. Cette approche est utile lorsque vous avez un grand nombre d'agents. Vous pouvez utiliser des balises de répartition des coûts pour organiser votre AWS facture afin de refléter votre propre structure de coûts. Pour ce faire, inscrivez-vous pour obtenir la facture de votre AWS compte incluant les clés et les valeurs des balises. Pour plus d'informations, consultez Configuration du rapport de répartition des coûts mensuel dans le Guide d'utilisateur AWS Billing .

Par exemple, vous pouvez ajouter des balises qui représentent le centre de coûts et l'objectif de vos ressources Amazon MQ :

Ressource	Clé	Valeur
`Broker1`	`Cost Center`	`34567`
	`Stack`	`Production`
`Broker2`	`Cost Center`	`34567`
	`Stack`	`Production`
`Broker3`	`Cost Center`	`12345`
	`Stack`	`Development`

Cette méthode de balisage vous permet de regrouper les deux agents effectuant les tâches connexes dans le même centre de coûts, tout en balisant un agent indépendant avec une autre balise de répartition des coûts.

Ajouter des balises dans la console Amazon MQ

Vous pouvez rapidement ajouter des balises aux ressources que vous créez dans la console Amazon MQ en suivant ces étapes :

1. Sur la page Create a broker (Créer un agent), sélectionnez Paramètres supplémentaires.
2. Sous Balises, sélectionnez Ajouter une balise.
3. Entrer une paire de Clés et de Valeurs.
4. (Facultatif) Sélectionnez Ajouter une balise pour ajouter plusieurs balises à votre agent.
5. Sélectionnez Create broker (Créer un agent).

Pour ajouter des balises lorsque vous créez une configuration :

1. Sur la page Create configuration (Créer une configuration), sélectionnez Avancé.
2. Sous Balises sur la page Create configuration (Créer une configuration), sélectionnez Ajouter une balise.
3. Entrer une paire de Clés et de Valeurs.
4. (Facultatif) Sélectionnez Ajouter une balise pour ajouter plusieurs balises à votre configuration.
5. Sélectionnez Create configuration (Créer une configuration).

Après avoir ajouté des balises, vous pouvez afficher, modifier et supprimer les balises de vos ressources dans la console Amazon MQ. Vous pouvez également consulter les balises de vos ressources à l'aide de l'API REST. Pour plus d'informations, consultez la référence d'API REST Amazon MQ.

Utilisation d'Amazon MQ pour ActiveMQ

Amazon MQ facilite la création d'un agent de messages avec les ressources de calcul et de stockage adaptées à vos besoins. Vous pouvez créer, gérer et supprimer des courtiers à l'aide de l' AWS Management Console API REST Amazon MQ ou du. AWS Command Line Interface

Les courtiers Amazon MQ pour ActiveMQ peuvent être déployés en tant que courtiers à instance unique ou en tant que courtiers actifs/de réserve. Pour les deux modes de déploiement, Amazon MQ offre une durabilité élevée en stockant ses données de manière redondante.

> **Note**
>
> Amazon MQ utilise Apache KahaDB comme magasin de données. D'autres magasins de données, tels que JDBC et LevelDB, ne sont pas pris en charge.

Vous pouvez accéder à vos agents via tout langage de programmation pris en charge par ActiveMQ et en activant explicitement TLS pour les protocoles suivants :

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

Pour en savoir plus sur Amazon MQ REST APIs, consultez le manuel Amazon MQ REST API Reference.

Amazon MQ pour les courtiers ActiveMQ

Qu'est-ce qu'un courtier Amazon MQ pour ActiveMQ ?

Un agent est un environnement d'agent de messages qui s'exécute sur Amazon MQ. Il constitue la composante de base d'Amazon MQ. La description combinée de la classe d'instance de courtier (`m5`)

et de la taille (`large`,`medium`) est appelée type d'instance de courtier (par exemple,`mq.m5.large`). Pour de plus amples informations, veuillez consulter Broker instance types.

- Un agent à instance unique est composé d'un agent dans une zone de disponibilité. L'agent communique avec votre application et avec un volume de stockage Amazon EBS ou Amazon EFS.
- Un agent actif/en veille est composé de deux agents répartis dans deux zones de disponibilité différentes, configurés dans une paire redondante. Ces agents communiquent de manière synchrone avec votre application et avec Amazon EFS.

Pour de plus amples informations, veuillez consulter Options de déploiement pour Amazon MQ pour les courtiers ActiveMQ.

Vous pouvez activer les mises à niveau automatiques des versions mineures vers de nouvelles versions mineures pour le moteur d'agent, à mesure qu'Apache publie de nouvelles versions. Les mises à niveau automatiques se produisent pendant la fenêtre de maintenance définie par le jour de la semaine, l'heure de la journée (au format 24 heures) et le fuseau horaire (UTC par défaut).

Pour plus d'informations sur la création et la gestion des agents, consultez les sections suivantes :

- Mise en route : création et connexion à un courtier ActiveMQ
- Agents
- Broker statuses

Protocoles de niveau filaire pris en charge

Vous pouvez accéder à vos agents via tout langage de programmation pris en charge par ActiveMQ et en activant explicitement TLS pour les protocoles suivants :

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

Attributs

Un agent ActiveMQ a plusieurs attributs, par exemple :

- Un nom (`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`)
- Une URL de console web ActiveMQ (`https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:8162`)

  Pour plus d'informations, consultez Console web dans la documentation ActiveMQ Apache.

  > **Important**
  >
  > Si vous spécifiez un plan d'autorisation qui n'inclut pas le groupe `activemq-webconsole`, vous ne pouvez pas utiliser la console web ActiveMQ car le groupe n'est pas autorisé à envoyer des messages à l'agent Amazon MQ ou à recevoir des messages de ce dernier.

- Des points de terminaison de protocole de niveau filaire:
  - `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**
    >
    > Il s'agit d'un OpenWire point final.

  - `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`

Pour plus d'informations, consultez Configuration des transports dans la documentation ActiveMQ Apache.

> **Note**
>
> Pour un active/standby courtier, Amazon MQ fournit deux consoles Web ActiveMQ URLs, mais une seule URL est active à la fois. De même, Amazon MQ fournit deux points de terminaison pour chaque protocole de niveau filaire, mais un seul point de terminaison est actif dans chaque paire à la fois. Les suffixes `-1` et `-2` indiquent une paire redondante.

Pour obtenir la liste complète des attributs des agents, consultez ce qui suit dans la référence d'API REST Amazon MQ :

- ID d'opération REST : Agent
- ID d'opération REST : Agents
- ID d'opération REST : Redémarrage d'agent

Utilisateurs du courtier

Un utilisateur ActiveMQ est une personne ou une application qui peut accéder aux files d'attente et aux rubriques d'un agent ActiveMQ. Vous pouvez configurer les utilisateurs pour qu'ils disposent d'autorisations spécifiques. Par exemple, vous pouvez autoriser certains utilisateurs à accéder à la console web ActiveMQ.

Un groupe est une étiquette sémantique. Vous pouvez affecter un groupe à un utilisateur et configurer des autorisations pour les groupes pour envoyer vers, recevoir depuis et administrer des files d'attente et des rubriques spécifiques.

> **Important**
>
> Apporter des modifications à une configuration n'applique pas immédiatement les modifications à l'agent. Pour appliquer vos modifications, vous devez attendre la fenêtre de maintenance suivante ou redémarrer l'agent.

Pour plus d'informations sur les utilisateurs et les groupes, consultez les éléments suivants dans la documentation Apache ActiveMQ :

- Autorisation
- Exemple d'autorisation

Pour plus d'informations sur la création, la modification et la suppression des utilisateurs ActiveMQ, consultez les sections suivantes :

- Création d'un utilisateur de courtier ActiveMQ
- Users

Attributs utilisateur

Pour obtenir la liste complète des attributs utilisateur, consultez les sections suivantes dans la référence des API REST Amazon MQ :

- ID d'opération REST : Utilisateur
- ID d'opération REST : Utilisateurs

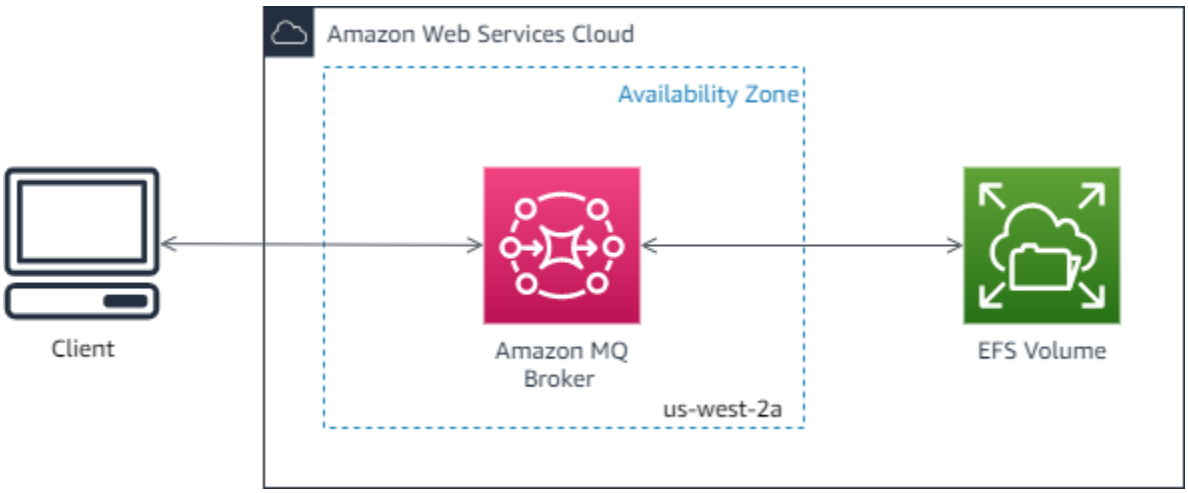
Options de déploiement pour Amazon MQ pour les courtiers ActiveMQ

Amazon MQ propose des options de déploiement en instance unique et en cluster pour les courtiers.

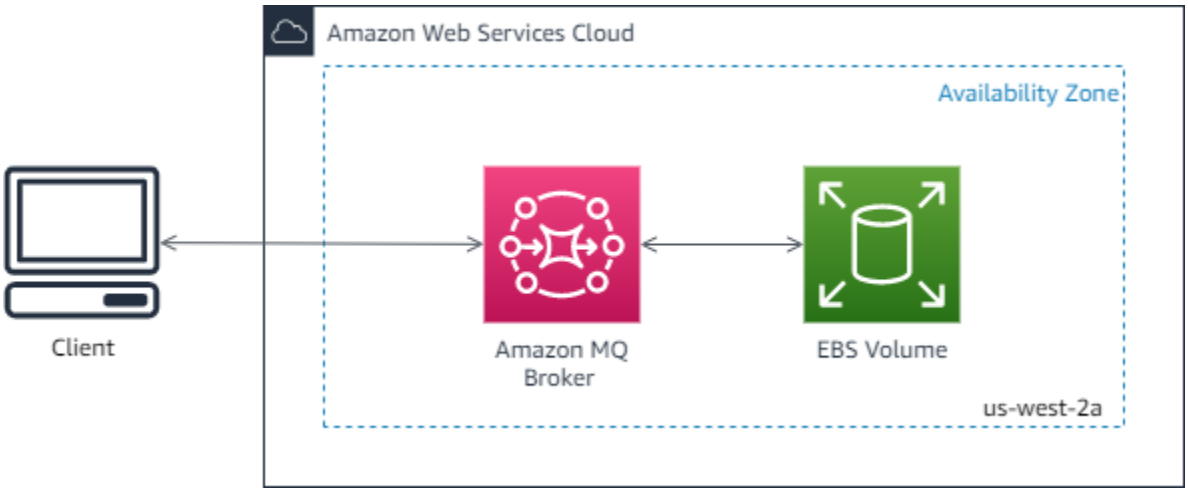
Option 1 : courtiers en instance unique Amazon MQ

Un agent à instance unique est composé d'un agent dans une zone de disponibilité. L'agent communique avec votre application et avec un volume de stockage Amazon EBS ou Amazon EFS. Les volumes de stockage Amazon EFS sont conçus pour fournir le plus haut niveau de durabilité et de disponibilité en stockant les données de manière redondante dans plusieurs zones de disponibilité ()AZs. Amazon EBS fournit un stockage de niveau bloc optimisé pour une faible latence et un débit élevé. Pour plus d'informations sur les options de stockage, consultez Storage.

Le schéma suivant illustre un broker à instance unique avec un stockage Amazon EFS répliqué sur plusieurs instances. AZs



Le diagramme suivant illustre un agent à instance unique avec un stockage Amazon EBS répliqué sur plusieurs serveurs dans une même zone de disponibilité.



Option 2 : active/standby courtiers Amazon MQ pour une haute disponibilité

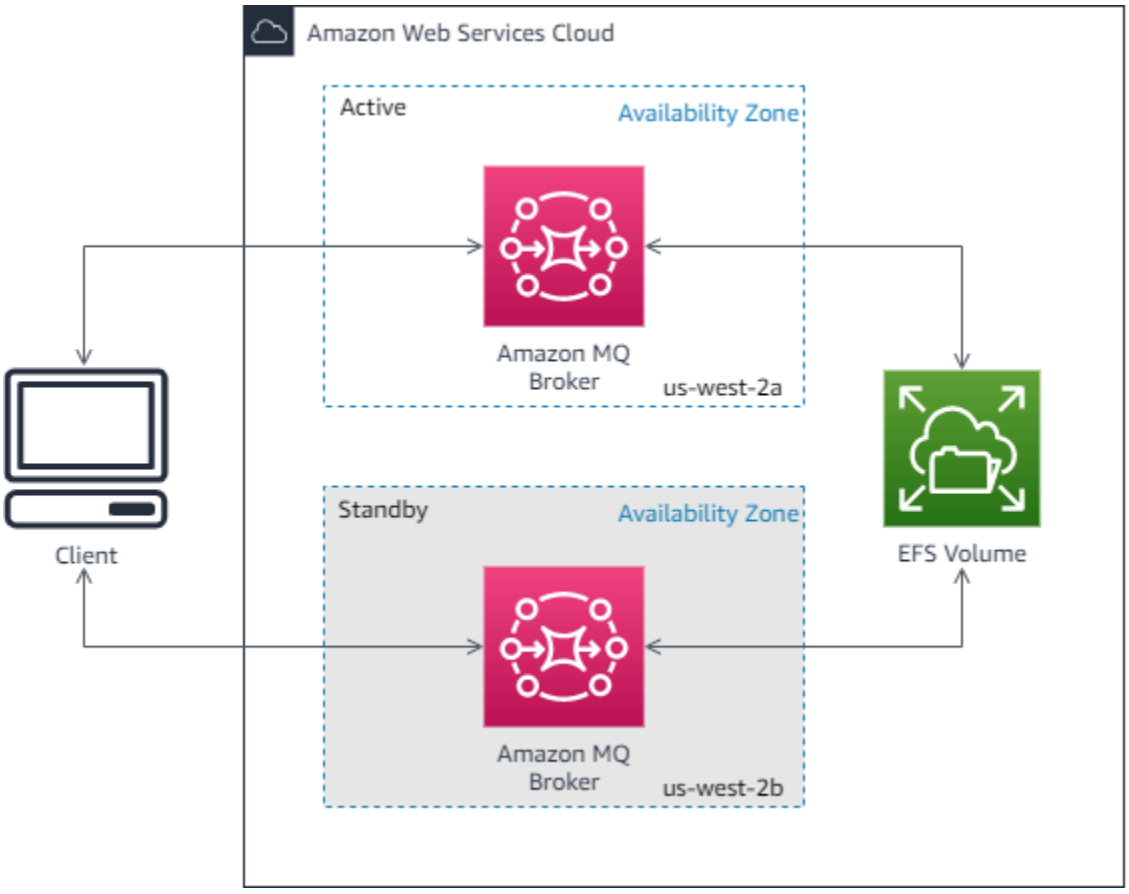
Un agent actif/en veille est composé de deux agents répartis dans deux zones de disponibilité différentes, configurés dans une paire redondante. Ces agents communiquent de manière synchrone avec votre application et avec Amazon EFS. Les volumes de stockage Amazon EFS sont conçus pour fournir le plus haut niveau de durabilité et de disponibilité en stockant les données de manière redondante dans plusieurs zones de disponibilité ()AZs. Pour de plus amples informations, veuillez consulter Storage.

Généralement, une seule des instances d'agent à la fois est active, les autres instances d'agent étant en veille. Si l'une des instances de l'agent est défaillante ou est en cours de maintenance, Amazon MQ met rapidement l'instance inactive hors service. Cela permet à l'instance en veille intègre de devenir active et de commencer à accepter les communications entrantes. Les fenêtres

de maintenance et les redémarrages des courtiers que vous initiez provoqueront un basculement. Lorsque vous redémarrez un agent, le basculement ne prend que quelques secondes.

Pour un active/standby courtier, Amazon MQ fournit deux consoles Web ActiveMQ URLs, mais une seule URL est active à la fois. De même, Amazon MQ fournit deux points de terminaison pour chaque protocole de niveau filaire, mais un seul point de terminaison est actif dans chaque paire à la fois. Les suffixes `-1` et `-2` indiquent une paire redondante. Pour les points de terminaison du protocole filaire, vous devez autoriser votre application à se connecter à l'un ou l'autre point de terminaison à l'aide du transport Failover.

Le schéma suivant illustre un active/standby broker dont le stockage Amazon EFS est répliqué sur plusieurs AZs sites.



Réseau de courtiers Amazon MQ

Amazon MQ prend en charge la fonction de réseau d'agents d'ActiveMQ.

Un réseau de courtiers est composé de plusieurs courtiers ou active/standby courtiers à instance unique actifs simultanément. La création d'un réseau de courtiers peut améliorer la disponibilité, la tolérance aux pannes et l'équilibrage de charge grâce à plusieurs instances de courtiers.

Comment fonctionne un réseau de courtiers ?

Un réseau de courtiers est établi en connectant un courtier à un autre à l'aide de connecteurs réseau. Un connecteur réseau fournit des messages à la demande d'un courtier à un autre. Les connecteurs réseau sont configurés dans la configuration du broker sous forme de connexions non duplex ou duplex. Avec des connexions non duplex, les messages sont transférés uniquement d'un agent à un autre. Pour les connexions duplex, les messages sont transférés dans les deux sens entre les deux courtiers.

Si le connecteur réseau est configuré en mode duplex, les messages sont également transférés de Broker2 à Broker1.

Vous pouvez utiliser à la fois des connexions duplex et non duplex dans un réseau de courtiers. Vous souhaiterez peut-être introduire une connexion duplex avec un autre courtier pour améliorer le trafic ou pour éviter une augmentation de limite. Les connexions duplex sont également utiles pour la migration partielle des courtiers sur site vers les courtiers gérés par Amazon MQ.

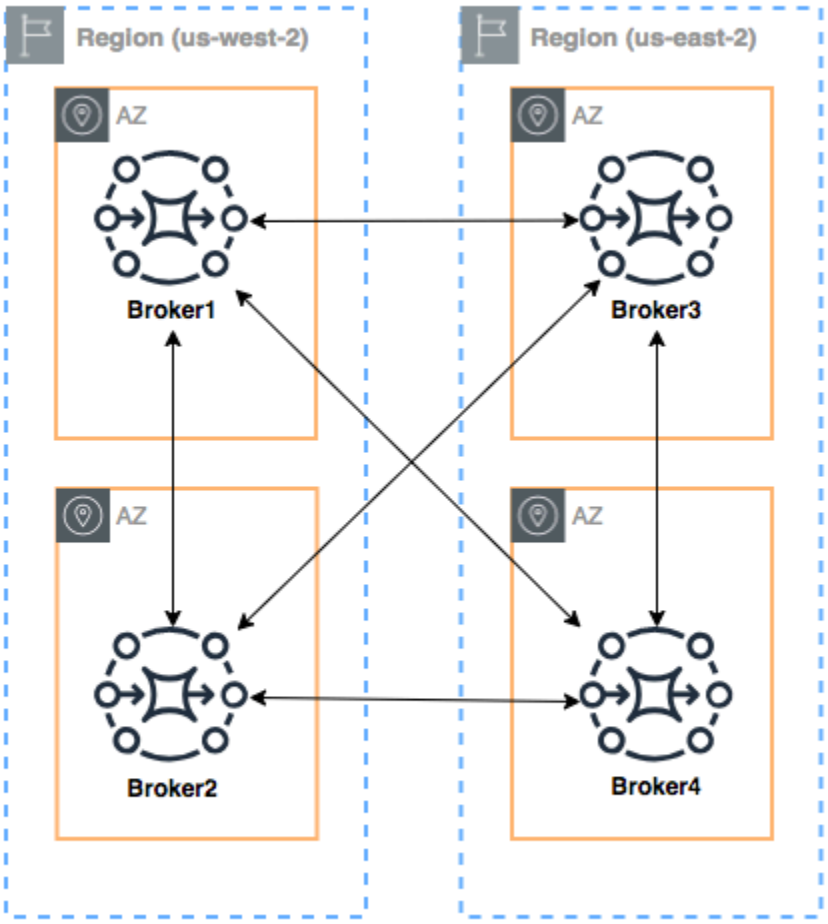
Comment un réseau d'agents gère-t-il les informations d'identification ?

Pour qu'un agent A se connecte à un agent B dans un réseau, l'agent A doit utiliser des informations d'identification valides, comme tout autre producteur ou consommateur. Au lieu de fournir un mot de passe dans la configuration `<networkConnector>` de l'agent A, vous devez d'abord créer un utilisateur sur l'agent A avec les mêmes valeurs qu'un autre utilisateur sur l'agent B (ceux-ci sont des utilisateurs séparés, uniques qui partagent les mêmes valeurs de nom d'utilisateur et de mot de passe). Lorsque vous spécifiez l'attribut `userName` dans la configuration `<networkConnector>`, Amazon MQ ajoute le mot de passe automatiquement lors de l'exécution.

> **Important**
>
> Ne spécifiez pas l'attribut `password` pour la configuration `<networkConnector>`. Nous vous déconseillons de stocker les mots de passe en texte brut dans les fichiers de configuration de l'agent, car les mots de passe deviennent visibles dans la console Amazon MQ. Pour de plus amples informations, veuillez consulter Configure Network Connectors for Your Broker.

Entre régions

Pour configurer un réseau de courtiers couvrant plusieurs AWS régions, déployez des courtiers dans ces régions et configurez des connecteurs réseau vers les points de terminaison de ces courtiers.



Pour configurer un réseau d'agents comme dans cet exemple, vous pouvez ajouter des entrées `networkConnectors` aux configurations des agents Broker1 et Broker4 qui référencent les points de terminaison de niveau filaire de ces agents.

Connecteurs de réseau pour 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>
```

Connecteur de réseau pour 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>
```

Connecteurs de réseau pour 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>
```

Basculement dynamique avec des connecteurs de transport

En plus de configurer les éléments `networkConnector`, vous pouvez configurer les options `transportConnector` d'agent pour activer le basculement dynamique et rééquilibrer les connexions lorsque des agents sont ajoutés ou supprimés du réseau.

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

Dans cet exemple, `updateClusterClients` et `rebalanceClusterClients` sont définis sur `true`. Dans ce cas, les clients recevront une liste d'agents dans le réseau et vous leur demanderez d'effectuer un rééquilibrage si un nouvel agent est ajouté.

Options disponibles :

- `updateClusterClients` : Transmet aux clients des informations sur les modifications apportées au réseau de topologie d'agent.
- `rebalanceClusterClients` : Entraîne le rééquilibrage des clients entre les agents lorsqu'un nouvel agent est ajouté à un réseau d'agents.
- `updateClusterClientsOnRemove` : Met à jour les clients avec des informations de topologie lorsqu'un agent quitte un réseau d'agents.

Quand `updateClusterClients` est défini sur true, les clients peuvent être configurés pour se connecter à un seul agent dans un réseau d'agents.

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

Lorsqu'un nouveau courtier se connecte, il reçoit une liste URIs de tous les courtiers du réseau. Si la connexion à l'agent échoue, un basculement dynamique peut être effectué sur l'un des agents fournis lorsque la connexion a été établie.

Pour plus d'informations sur le basculement, consultez Broker-side Options for Failover dans la documentation Active MQ.

Types d'instances de courtier Amazon MQ pour ActiveMQ

La description combinée de la classe d'instance de courtier (`m5`) et de la taille (`large`,`medium`) est appelée type d'instance de courtier (par exemple,`mq.m5.large`). Le tableau suivant répertorie les types d'instances de courtier Amazon MQ disponibles pour les courtiers ActiveMQ.

Amazon MQ fournit un préavis d'au moins 90 jours avant la fin du support d'un type d'instance. Nous vous recommandons de mettre à niveau votre courtier vers un nouveau type d'instance avant end-of-support cette date afin d'éviter toute interruption.

> **Important**
>
> Vous ne pouvez pas créer de `t2.micro` courtiers à `mq.m4.large` compter du 17 mars 2025.

Type d'instance	vCPU	Mémoire (Gio)	Utilisation recommandée	Stockage	Fin du support sur Amazon MQ
`mq.t3.micro`	2	1	Evaluation	EFS	
`mq.m5.large`	2	8	Production	EFS ou EBS	
`mq.m5.xlarge`	4	16	Production	EFS ou EBS	
`mq.m5.2xlarge`	8	32	Production	EFS ou EBS	
`mq.m5.4xlarge`	16	64	Production	EFS ou EBS	

Pour plus d'informations sur le débit, consultez Choisir le type d'instance d'agent adéquat pour un débit optimal.

Configurations d'agent Amazon MQ for ActiveMQ

Une configuration contient tous les paramètres de votre agent ActiveMQ au format XML (à l'instar du fichier `activemq.xml` d'ActiveMQ). Vous pouvez créer une configuration avant de créer des agents. Vous pouvez ensuite appliquer la configuration à un ou plusieurs agents.

> **Important**
>
> Apporter des modifications à une configuration n'applique pas immédiatement les modifications à l'agent. Pour appliquer vos modifications, vous devez attendre la fenêtre de maintenance suivante ou redémarrer l'agent.
>
> Vous ne pouvez supprimer une configuration qu'à l'aide de l'`DeleteConfiguration`API. Pour plus d'informations, consultez Configurations dans le manuel Amazon MQ API Reference.

Attributes

La configuration d'un agent a plusieurs attributs, par exemple :

- Un nom (`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`)

Pour obtenir la liste complète des attributs de configuration, consultez ce qui suit dans la référence d'API REST Amazon MQ :

- ID d'opération REST : Configuration
- ID d'opération REST : Configurations

Pour obtenir la liste complète des attributs des révisions de configuration, consultez les sections suivantes :

- ID d'opération REST : Révision de configuration
- ID d'opération REST : Révisions de configuration

Utilisation des fichiers de configuration XML Spring

Les agents ActiveMQ sont configurés à l'aide de fichiers XML Spring. Vous pouvez configurer de nombreux aspects de votre agent ActiveMQ, comme les destinations prédéfinies, les politiques de destination, les politiques d'autorisation et les plugins. Amazon MQ contrôle certains de ces éléments de configuration, tels que les transports et le stockage réseau. D'autres options de configuration, telles que la création de réseaux d'agents, ne sont pas prises en charge actuellement.

L'ensemble complet des options de configuration prises en charge est spécifié dans les schémas XML Amazon MQ. Téléchargez les fichiers zip des schémas pris en charge en cliquant sur les liens suivants.

- `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

Vous pouvez utiliser ces schémas pour valider et nettoyer vos fichiers de configuration. Amazon MQ vous permet également de fournir des configurations en chargeant des fichiers XML. Lorsque vous chargez un fichier XML, Amazon MQ supprime et nettoie automatiquement les paramètres de configuration non valides et interdits selon le schéma.

> **Note**
>
> Vous pouvez uniquement utiliser des valeurs statiques pour les attributs. Amazon MQ nettoie les éléments et attributs qui contiennent des variables, des références d'élément et des expressions Spring.

Création d'une configuration de broker Amazon MQ pour ActiveMQ

Une configuration contient tous les paramètres de votre agent ActiveMQ au format XML (similaire au fichier `activemq.xml` d'ActiveMQ). Vous pouvez créer une configuration avant de créer des agents. Vous pouvez ensuite appliquer la configuration à un ou plusieurs agents. Vous pouvez appliquer une configuration immédiatement ou au cours d'une fenêtre de maintenance.

L'exemple suivant montre comment créer et appliquer une configuration d'agent Amazon MQ à l'aide de AWS Management Console.

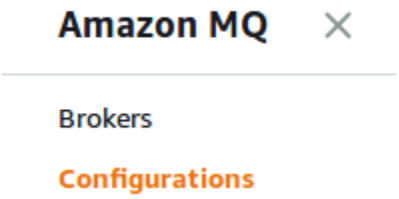
> **Important**
>
> Vous ne pouvez supprimer une configuration qu'à l'aide de l'`DeleteConfiguration`API. Pour plus d'informations, consultez Configurations dans le manuel Amazon MQ API Reference.

Création d'une nouvelle configuration

Pour créer une nouvelle configuration de broker, créez d'abord la nouvelle configuration.

1. Connectez-vous à la console Amazon MQ.

2. Sur la gauche, développez le volet de navigation et choisissez Configurations.



3. Sur la page Configurations, choisissez Create configuration (Créer une configuration).
4. Sur la page Create configuration (Créer une configuration), dans la section Details (Détails), saisissez le Configuration name (Nom de configuration) (par exemple, `MyConfiguration`) et sélectionnez une version de Broker engine (Moteur d'agent).

> **Note**
>
> Pour en savoir plus sur les versions de moteur ActiveMQ prises en charge par Amazon MQ for ActiveMQ, consultez the section called “Gestion des versions”.

5. Choisissez Créer une configuration.

Créer une révision de configuration

Après avoir créé une configuration de broker, vous devez la modifier à l'aide d'une révision de configuration.

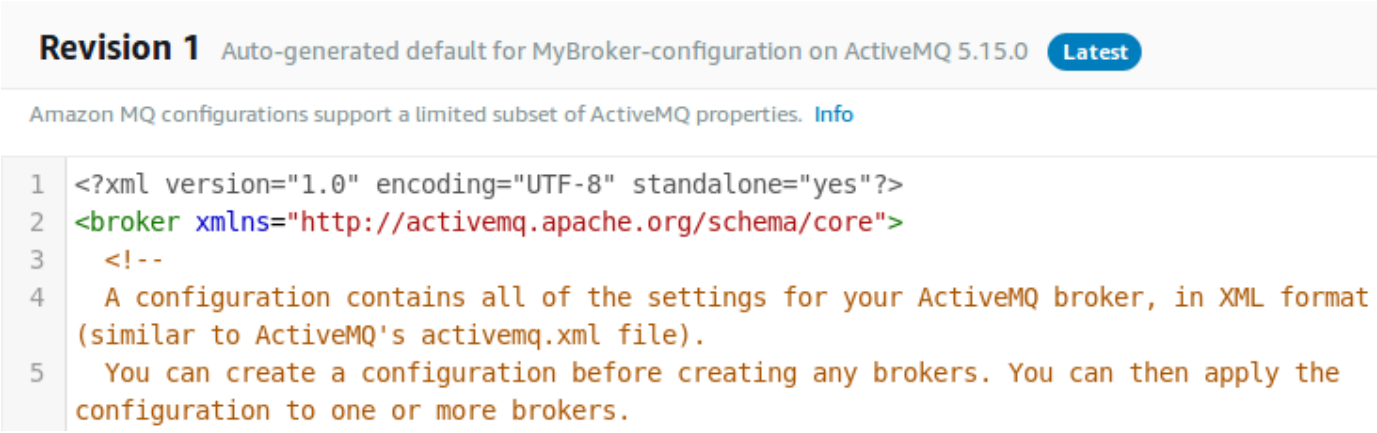
1. Dans la liste des configurations, choisissez ***`MyConfiguration`***.

> **Note**
>
> La première révision de configuration est toujours créée lorsqu'Amazon MQ crée la configuration.

Sur la ***`MyConfiguration`***page, le type et la version du moteur de courtage utilisés par votre nouvelle révision de configuration (par exemple, Apache ActiveMQ 5.15.16) sont affichés.

2. Dans l'onglet Configuration details, le numéro de révision de configuration, la description et la configuration d'agent au format XML sont affichés.

> **Note**
>
> La modification de la configuration actuelle crée une nouvelle révision de configuration.



3. Choisissez Edit configuration (Modifier la configuration) et apportez des modifications à la configuration XML.
4. Choisissez Enregistrer.

   La boîte de dialogue Save revision (Enregistrer la révision) s'affiche.
5. (Facultatif) Type `A description of the changes in this revision`.
6. Choisissez Save (Enregistrer).

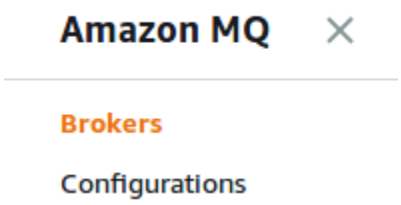
   La nouvelle révision de configuration est enregistrée.

> **Important**
>
> La console Amazon MQ nettoie automatiquement les paramètres de configuration non valides et interdits selon un schéma. Pour plus d'informations et une liste complète des paramètres XML autorisés, consultez Amazon MQ Broker Configuration Parameters.

Appliquer une révision de configuration à votre agent

Après avoir révisé la configuration, vous pouvez appliquer la révision de configuration à votre courtier.

1. Sur la gauche, développez le volet de navigation et choisissez Brokers (Agents).



2. Dans la liste des courtiers, sélectionnez votre courtier (par exemple MyBroker), puis choisissez Modifier.
3. Sur la *MyBroker* page Modifier, dans la section Configuration, sélectionnez une configuration et une révision, puis choisissez Planifier les modifications.
4. Dans la section Schedule broker modifications (Planifier les modifications de l'agent), choisissez si les modifications doivent être appliquées During the next scheduled maintenance window (Au cours de la prochaine fenêtre de maintenance) ou Immediately (immédiatement).

> **Important**
>
> Les courtiers à instance unique sont hors ligne lors du redémarrage. Pour les courtiers en clusters, un seul nœud est en panne à la fois lorsque le courtier redémarre.

5. Cliquez sur Appliquer.

   Votre révision de configuration est appliquée à votre agent à l'heure spécifiée.

Modifier une révision de configuration Amazon MQ pour ActiveMQ

Vous souhaiterez peut-être modifier une révision de configuration après l'avoir appliquée à votre courtier. Suivez les instructions ci-dessous pour modifier une révision de configuration.

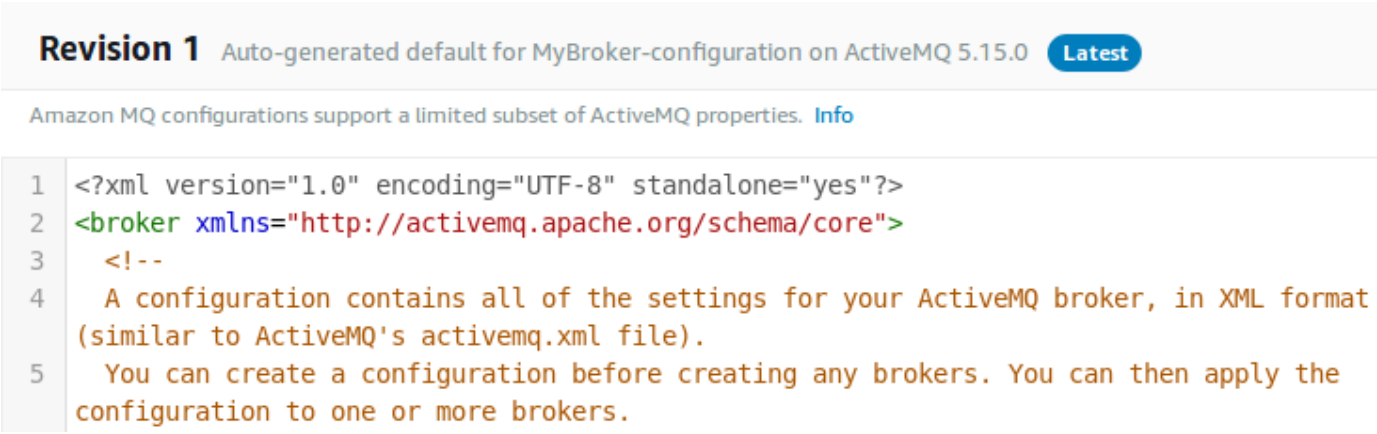
1. Connectez-vous à la console Amazon MQ.
2. Dans la liste des courtiers, sélectionnez votre courtier (par exemple MyBroker), puis choisissez Modifier.
3. Sur la ***MyBroker***page, choisissez Modifier.
4. Sur la *MyBroker* page Modifier, dans la section Configuration, sélectionnez une configuration et une révision, puis choisissez Modifier.

> **Note**
>
> Sauf si vous sélectionnez une configuration lorsque vous créez un agent, la première révision de configuration est toujours créée lorsqu'Amazon MQ crée l'agent.

Sur la ***MyBroker***page, le type et la version du moteur de courtage utilisés par la configuration (par exemple, Apache ActiveMQ 5.15.8) sont affichés.

5. Dans l'onglet Configuration details, le numéro de révision de configuration, la description et la configuration d'agent au format XML sont affichés.

> **Note**
>
> La modification de la configuration actuelle crée une nouvelle révision de configuration.



6. Choisissez Edit configuration (Modifier la configuration) et apportez des modifications à la configuration XML.
7. Choisissez Enregistrer.

   La boîte de dialogue Save revision (Enregistrer la révision) s'affiche.
8. (Facultatif) Type `A description of the changes in this revision`.
9. Choisissez Save (Enregistrer).

   La nouvelle révision de configuration est enregistrée.

> **Important**
>
> La console Amazon MQ nettoie automatiquement les paramètres de configuration non valides et interdits selon un schéma. Pour plus d'informations et une liste complète des paramètres XML autorisés, consultez Amazon MQ Broker Configuration Parameters.

Éléments autorisés dans les configurations Amazon MQ

Voici une liste détaillée des éléments autorisés dans les configurations Amazon MQ. Pour plus d'informations, consultez la page Configuration XML dans la documentation ActiveMQ Apache.

Element
`abortSlowAckConsumerStrategy` (attributs)
`abortSlowConsumerStrategy` (attributs)
`authorizationEntry` (attributs)
`authorizationMap` (Éléments de collection enfant)
`authorizationPlugin` (Éléments de collection enfant)
`broker` (attributs | Éléments de collection enfant)
`cachedMessageGroupMapFactory` (attributs)
`compositeQueue` (attributs | Éléments de collection enfant)
`compositeTopic` (attributs | Éléments de collection enfant)
`constantPendingMessageLimitStrategy` (attributs)
`discarding` (attributs)
`discardingDLQBrokerPlugin` (attributs)
`fileCursor`

Element
`fileDurableSubscriberCursor`
`fileQueueCursor`
`filteredDestination` (attributs)
`fixedCountSubscriptionRecoveryPolicy` (attributs)
`fixedSizedSubscriptionRecoveryPolicy` (attributs)
`forcePersistencyModeBrokerPlugin` (attributs)
`individualDeadLetterStrategy` (attributs)
`lastImageSubscriptionRecoveryPolicy`
`messageGroupHashBucketFactory` (attributs)
`mirroredQueue` (attributs)
`noSubscriptionRecoveryPolicy`
`oldestMessageEvictionStrategy` (attributs)
`oldestMessageWithLowestPriorityEvictionStrategy` (attributs)
`policyEntry` (attributs | Éléments de collection enfant)
`policyMap` (Éléments de collection enfant)
`prefetchRatePendingMessageLimitStrategy` (attributs)
`priorityDispatchPolicy`
`priorityNetworkDispatchPolicy`
`queryBasedSubscriptionRecoveryPolicy` (attributs)
`queue` (attributs)

Element
`redeliveryPlugin` (attributs | Éléments de collection enfant)
`redeliveryPolicy` (attributs)
`redeliveryPolicyMap` (Éléments de collection enfant)
`retainedMessageSubscriptionRecoveryPolicy` (Éléments de collection enfant)
`roundRobinDispatchPolicy`
`sharedDeadLetterStrategy` (attributs | Éléments de collection enfant)
`simpleDispatchPolicy`
`simpleMessageGroupMapFactory`
`statisticsBrokerPlugin`
`storeCursor`
`storeDurableSubscriberCursor` (attributs)
`strictOrderDispatchPolicy`
`tempDestinationAuthorizationEntry` (attributs)
`tempQueue` (attributs)
`tempTopic` (attributs)
`timedSubscriptionRecoveryPolicy` (attributs)
`timeStampingBrokerPlugin` (attributs)
`topic` (attributs)
`transportConnector` (attributs)
`uniquePropertyMessageEvictionStrategy` (attributs)

Element	
virtualDestinationInterceptor	(Éléments de collection enfant)
virtualTopic	(attributs)
vmCursor	
vmDurableCursor	
vmQueueCursor	

Éléments et leurs attributs autorisés dans les configurations Amazon MQ

Voici une liste détaillée des éléments et de leurs attributs autorisés dans les configurations Amazon MQ. Pour plus d'informations, consultez la page Configuration XML dans la documentation ActiveMQ Apache.

Element	Attribut
abortSlowAckConsumerStrategy	abortConnection
	checkPeriod
	ignoreIdleConsumers
	ignoreNetworkConsumers
	maxSlowCount
	maxSlowDuration
	maxTimeSinceLastAck
	name
abortSlowConsumerStrategy	abortConnection
	checkPeriod

Element	Attribut
	ignoreNetworkConsumers
	maxSlowCount
	maxSlowDuration
	name
authorizationEntry	admin
	queue
	read
	tempQueue
	tempTopic
	topic
	write
broker	advisorySupport
	allowTempAutoCreationOnSend
	cacheTempDestinations
	consumerSystemUsagePortion
	dedicatedTaskRunner
	deleteAllMessagesOnStartup
	keepDurableSubsActive
	enableMessageExpirationOnActiveDurableSubs
	maxPurgedDestinationsPerSweep

Element	Attribut
	maxSchedulerRepeatAllowed
	monitorConnectionSplits
	networkConnectorStartAsync
	offlineDurableSubscriberTaskSchedule
	offlineDurableSubscriberTimeout
	persistenceThreadPriority
	persistent
	populateJMSXUserID
	producerSystemUsagePortion
	rejectDurableConsumers
	rollbackOnlyOnAsyncException
	schedulePeriodForDestinationPurge
	schedulerSupport
	splitSystemUsageForProducersConsumers
	taskRunnerPriority
	timeBeforePurgeTempDestinations
	useAuthenticatedPrincipalForJMSXUserID
	useMirroredQueues

Element	Attribut
	useTempMirroredQueues
	useVirtualDestSubs
	useVirtualDestSubsOnCreation
	useVirtualTopics
cachedMessageGroupMapFactory	cacheSize
compositeQueue	concurrentSend
	copyMessage
	forwardOnly
	name
	sendWhenNotMatched
compositeTopic	concurrentSend
	copyMessage
	forwardOnly
	name
	sendWhenNotMatched
conditionalNetworkBridgeFilterFactory	rateDuration
	rateLimit
	replayDelay
	replayWhenNoConsumers

Element	Attribut
	`selectorAware` Pris en charge dans Apache ActiveMQ 15.16.x
`constantPendingMessageLimitStrategy`	`limit`
`discarding`	`deadLetterQueue`
	`enableAudit`
	`expiration`
	`maxAuditDepth`
	`maxProducersToAudit`
	`processExpired`
	`processNonPersistent`
`discardingDLQBrokerPlugin`	`dropAll`
	`dropOnly`
	`dropTemporaryQueues`
	`dropTemporaryTopics`
	`reportInterval`
`filteredDestination`	`queue`
	`selector`
	`topic`

Element	Attribut
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

Element	Attribut
	postfix
	prefix
oldestMessageEvictionStrategy	evictExpiredMessagesHighWatermark
oldestMessageWithLowestPriorityEvictionStrategy	evictExpiredMessagesHighWatermark
policyEntry	advisoryForConsumed
	advisoryForDelivery
	advisoryForDiscardingMessages
	advisoryForFastProducers
	advisoryForSlowConsumers
	advisoryWhenFull
	allConsumersExclusiveByDefault
	alwaysRetroactive
	blockedProducerWarningInterval
	consumersBeforeDispatchStarts
	cursorMemoryHighWaterMark
	doOptimzeMessageStorage
	durableTopicPrefetch
	enableAudit
	expireMessagesPeriod

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

Element	Attribut
	`producerFlowControl`
	`queue`
	`queueBrowserPrefetch`
	`queuePrefetch`
	`reduceMemoryFootprint`
	`sendAdvisoryIfNoConsumers`
	`sendFailIfNoSpace`
	`sendFailIfNoSpaceAfterTimeout` Pris en charge dans Apache ActiveMQ 15.16.4 et versions ultérieures
	`sendDuplicateFromStoreToDLQ`
	`storeUsageHighWaterMark`
	`strictOrderDispatch`
	`tempQueue`
	`tempTopic`
	`timeBeforeDispatchStarts`
	`topic`
	`topicPrefetch`
	`useCache`

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

Element	Attribut
	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

Element	Attribut
	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

Element	Attribut
	`selectorAware`
	`setOriginalDestination`
	`transactedSend`

Attributs d'élément parent Amazon MQ

Voici une explication détaillée des attributs d'élément parent. Pour plus d'informations, consultez la page Configuration XML dans la documentation ActiveMQ Apache.

Rubriques

- agent

agent

`broker` est un élément de collecte parent.

Attributes

networkConnectionStartAsynchrone

Pour atténuer la latence du réseau et autoriser d'autres réseaux à démarrer en temps opportun, utilisez la balise `<networkConnectionStartAsync>`. La balise demande à l'agent d'utiliser un exécuteur pour démarrer des connexions réseau en parallèle et asynchrones au commencement d'un agent.

Par défaut : `false`

Exemple de configuration

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

Éléments, éléments de collection enfant et leurs éléments enfants autorisés dans les configurations Amazon MQ

Voici une liste détaillée des éléments, des éléments de collection enfant et de leurs éléments enfant autorisés dans les configurations Amazon MQ. Pour plus d'informations, consultez la page Configuration XML dans la documentation ActiveMQ Apache.

Element	Élément de collection enfant	Élément enfant
`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`

Element	Élément de collection enfant	Élément enfant
	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

Element	Élément de collection enfant	Élément enfant
compositeTopic	forwardTo	queue
		tempQueue
		tempTopic
		topic
		filteredDestination
policyEntry	deadLetterStrategy	discarding
		individualDeadLetterStrategy
		sharedDeadLetterStrategy
	destination	queue
		tempQueue
		tempTopic
		topic
	dispatchPolicy	priorityDispatchPolicy
		priorityNetworkDispatchPolicy
		roundRobinDispatchPolicy
		simpleDispatchPolicy
		strictOrderDispatchPolicy

Element	Élément de collection enfant	Élément enfant
		clientIdFilterDispatchPolicy
	messageEvictionStrategy	oldestMessageEvictionStrategy
		oldestMessageWithLowestPriorityEvictionStrategy
		uniquePropertyMessageEvictionStrategy
	messageGroupMapFactory	cachedMessageGroupMapFactory
		messageGroupHashBucketFactory
		simpleMessageGroupMapFactory
	pendingDurableSubscriberPolicy	fileDurableSubscriberCursor
		storeDurableSubscriberCursor
		vmDurableCursor
	pendingMessageLimitStrategy	constantPendingMessageLimitStrategy
		prefetchRatePendingMessageLimitStrategy
	pendingQueuePolicy	fileQueueCursor

Element	Élément de collection enfant	Élément enfant
		storeCursor
		vmQueueCursor
	pendingSubscriberPolicy	fileCursor
		vmCursor
	slowConsumerStrategy	abortSlowAckConsumerStrategy
		abortSlowConsumerStrategy
	subscriptionRecoveryPolicy	fixedCountSubscriptionRecoveryPolicy
		fixedSizedSubscriptionRecoveryPolicy
		lastImageSubscriptionRecoveryPolicy
		noSubscriptionRecoveryPolicy
		queryBasedSubscriptionRecoveryPolicy
		retainedMessageSubscriptionRecoveryPolicy
timedSubscriptionRecoveryPolicy		
policyMap	defaultEntry	policyEntry

Element	Élément de collection enfant	Élément enfant
	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

Element	Élément de collection enfant	Élément enfant
`virtualDestination Interceptor`	`virtualDestinations`	`compositeQueue`
		`compositeTopic`
		`virtualTopic`

Attributs d'élément enfant Amazon MQ

Voici une explication détaillée des attributs d'élément enfant. Pour plus d'informations, consultez la page Configuration XML dans la documentation ActiveMQ Apache.

Rubriques

- authorizationEntry
- networkConnector
- kahaDB
- systemUsage

authorizationEntry

`authorizationEntry` est un enfant de la collection d'élément enfant `authorizationEntries`.

Attributes

admin|read|write

Les autorisations accordées à un groupe d'utilisateurs. Pour de plus amples informations, veuillez consulter Toujours configurer un plan d'autorisation.

Si vous spécifiez un plan d'autorisation qui n'inclut pas le groupe `activemq-webconsole`, vous ne pouvez pas utiliser la console web ActiveMQ car le groupe n'est pas autorisé à envoyer des messages à l'agent Amazon MQ ou à recevoir des messages de ce dernier.

Par défaut : `null`

Exemple de configuration

```
<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**
>
> Le `activemq-webconsole` groupe dans ActiveMQ sur Amazon MQ possède des autorisations d'administrateur sur toutes les files d'attente et sur tous les sujets. Tous les utilisateurs de ce groupe auront un accès administrateur.

networkConnector

`networkConnector` est un enfant de la collection d'élément enfant `networkConnectors`.

Rubriques

- Attributes
- Exemples de configuration

Attributes

conduitSubscriptions

Indique si une connexion réseau dans un réseau d'agents traite plusieurs consommateurs abonnés à la même destination comme un seul consommateur. Par exemple, si `conduitSubscriptions` est défini comme `true` et que deux consommateurs se connectent à l'agent B et consomment à partir d'une destination, l'agent B combine les abonnements en un seul abonnement logique sur la connexion réseau de l'agent A, afin qu'une seule copie d'un message soit transférée de l'agent A à l'agent B.

> **Note**
>
> Définir `conduitSubscriptions` comme `true` peut réduire le trafic réseau redondant. Cependant, utiliser cet attribut peut avoir des conséquences pour l'équilibrage de charge des messages entre des consommateurs et cela peut entraîner un comportement incorrect dans certains scénarios (par exemple, avec des sélecteurs de messages JMS ou avec des rubriques durables).

Par défaut : `true`

duplex

Indique si la connexion dans le réseau d'agents est utilisée pour produire et consommer des messages. Par exemple, si l'agent A crée une connexion avec l'agent B en mode non duplex, les messages peuvent être uniquement transférés de l'agent A vers l'agent B. Toutefois, si l'agent A crée une connexion en duplex vers l'agent B, l'agent B peut alors transférer des messages vers l'agent A sans avoir à configurer de `<networkConnector>`.

Par défaut : `false`

name

Le nom du pont dans le réseau d'agents.

Par défaut : `bridge`

uri

Le point de terminaison de protocole de niveau filaire pour l'un des deux agents (ou pour plusieurs agents) dans un réseau d'agents.

Par défaut : `null`

nom d’utilisateur

Le nom d'utilisateur commun aux agents dans un réseau d'agents.

Par défaut : `null`

Exemples de configuration

> **Note**
>
> Lorsque vous utilisez un `networkConnector` pour définir un réseau d'agents, n'incluez pas le mot de passe pour l'utilisateur commun à vos agents.

Un réseau d'agents avec deux agents

Dans cette configuration, les deux agents sont connectés dans un réseau d'agents. Le nom du connecteur réseau est`connector_1_to_2`, le nom d'utilisateur commun aux courtiers est`myCommonUser`, la connexion est`duplex`, et l'URI du OpenWire point de terminaison est préfixé par`static:`, indiquant une one-to-one connexion entre les courtiers.

```
<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>
```

Pour de plus amples informations, veuillez consulter Configure Network Connectors for Your Broker.

Un réseau d'agents avec plusieurs agents

Dans cette configuration, plusieurs agents sont connectés dans un réseau d'agents. Le nom du connecteur réseau est`connector_1_to_2`, le nom d'utilisateur commun aux courtiers est`myCommonUser`, la connexion est`duplex`, et la liste des points de OpenWire terminaison séparés par des virgules URIs est préfixée par`masterslave:`, indiquant une connexion de basculement entre les courtiers. Le basculement d'un agent à un autre n'est pas aléatoire et les tentatives de reconnexion continuent indéfiniment.

```
<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**
>
> Nous vous recommandons d'utiliser le préfixe `masterslave:` pour le réseau d'agents. Le préfixe est identique à la syntaxe `static:failover:()?randomize=false&maxReconnectAttempts=0` plus explicite.

> **Note**
>
> Cette configuration XML n'autorise pas les espaces.

kahaDB

`kahaDB` est un enfant de la collection d'élément enfant `persistenceAdapter`.

Attributes

concurrentStoreAndDispatchQueues

Indique s'il convient d'utiliser la répartition et le stockage simultanés pour les files d'attente. Pour de plus amples informations, veuillez consulter Désactiver Concurrent Store and Dispatch (Répartition et stockage simultanés) pour les files d'attente à consommateurs lents.

Par défaut : `true`

cleanupOnStop

> **Pris en charge dans**
>
> Apache ActiveMQ 15.16.x et versions ultérieures

Si elle est désactivée, le récupérateur de mémoire et le nettoyage n'ont pas lieu lorsque l'agent est arrêté, ce qui accélère le processus d'arrêt. La vitesse accrue est utile dans les cas avec des bases de données volumineuses ou des bases de données de planificateur.

Par défaut : `true`

journalDiskSyncIntervalle

Intervalle (ms) indiquant quand effectuer une synchronisation de disque si `journalDiskSyncStrategy=periodic`. Pour de plus amples informations, veuillez consulter la documentation Apache ActiveMQ KahadB.

Par défaut : `1000`

journalDiskSyncStratégie

> Pris en charge dans
>
> Apache ActiveMQ 15.14.x et versions ultérieures

Configure la politique de synchronisation du disque. Pour de plus amples informations, veuillez consulter la documentation Apache ActiveMQ KahadB.

Par défaut : `always`

> Note
>
> La documentation ActiveMQ indique que la perte de données est limitée à la durée de `journalDiskSyncInterval`, qui a une valeur par défaut de 1 s. La perte de données peut être plus longue que l'intervalle, mais il est difficile d'être précis. Soyez prudent.

preallocationStrategy

Configure la façon dont l'agent va essayer de préallouer les fichiers journaux lorsqu'un nouveau fichier journal est nécessaire. Pour de plus amples informations, veuillez consulter la documentation Apache ActiveMQ KahadB.

Par défaut : `sparse_file`

Exemple de configuration

Example

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

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

systemUsage

`systemUsage` est un enfant de la collection d'élément enfant `systemUsage`. Il contrôle la quantité maximale d'espace que l'agent utilisera avant de ralentir les producteurs. Pour de plus amples informations, veuillez consulter Producer Flow Control dans la documentation Apache ActiveMQ.

Élément enfant

memoryUsage

`memoryUsage` est un enfant de l'élément enfant `systemUsage`. Il gère l'utilisation de la mémoire. Utilisez `memoryUsage` pour conserver une trace de la quantité d'utilisation d'un élément afin que vous puissiez contrôler efficacement l'utilisation de l'ensemble de travail. Pour de plus amples informations, veuillez consulter le schéma dans la documentation ActiveMQ Apache.

Élément enfant

`memoryUsage` est un enfant de l'élément enfant `memoryUsage`.

Attribut

percentOfJvmTas

Entier compris entre 0 (inclus) et 70 (inclus).

Par défaut : `70`

Attributes

sendFailIfNoSpace

Définit si une méthode `send()` doit échouer s'il n'y a pas d'espace libre. La valeur par défaut est false, ce qui bloque la méthode `send()` jusqu'à ce qu'il y ait de l'espace disponible. Pour de plus amples informations, veuillez consulter le schéma dans la documentation Apache Active MQ.

Par défaut : `false`

sendFailIfNoSpaceAfterTimeout

Par défaut : `null`

Exemple de configuration

Example

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

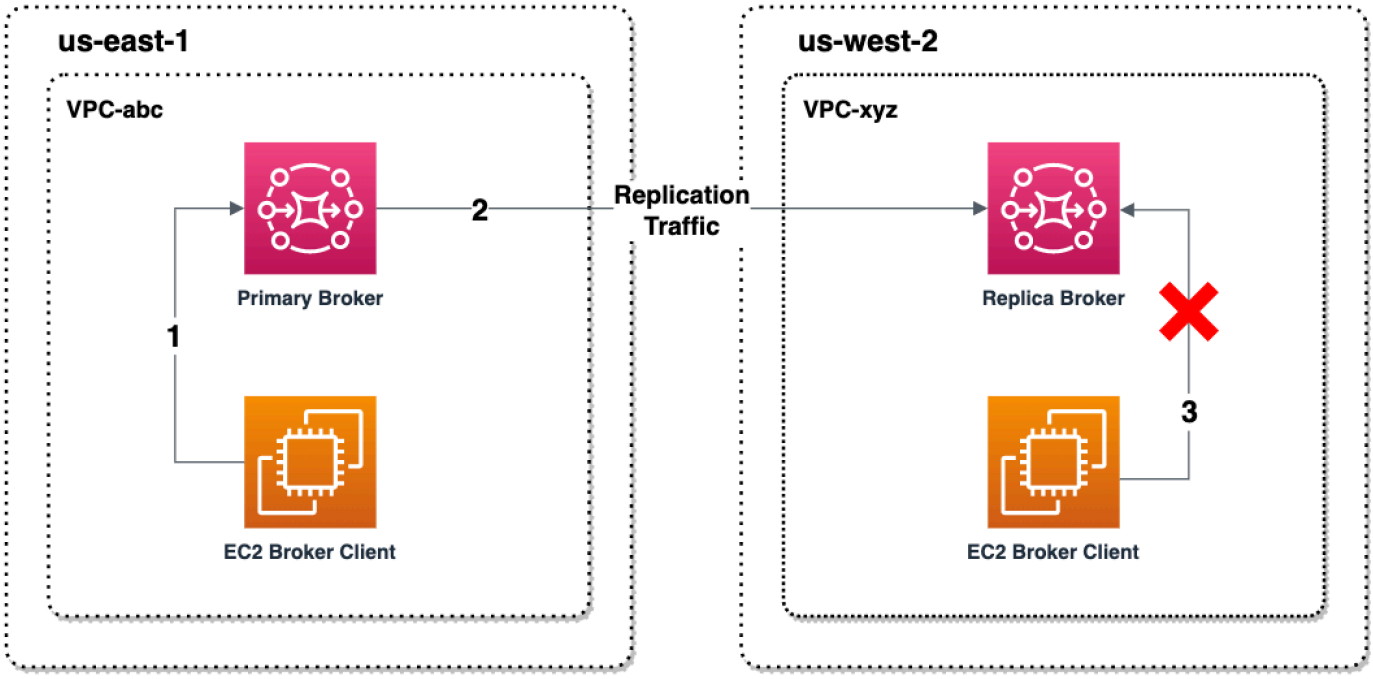
Réplication de données entre régions pour Amazon MQ pour ActiveMQ

Amazon MQ pour ActiveMQ propose une fonctionnalité de réplication des données entre régions (CRDR) qui permet la réplication asynchrone des messages depuis le courtier principal d'une région principale vers le courtier répliqué d'une région de réplication. AWS En émettant une demande de basculement à l'API Amazon MQ, l'agent de répliques actuel est promu au rôle d'agent principal et l'agent principal actuel est rétrogradé au rôle de réplique.

Brokers principaux et répliques pour la réplication de données entre régions

Vous pouvez créer des courtiers principaux et de répliques pour la réplication asynchrone des données depuis le courtier principal d'une AWS région principale vers le courtier de réplication d'une région de réplication. La région principale se compose d'une paire redondante d'agents actifs/en veille, appelée agent principal. La région secondaire se compose d'une paire redondante d'agents actifs/en veille, appelée agent de répliques.

Le schéma suivant illustre un agent de répliques dans une région secondaire recevant des données répliquées asynchrones de la part de l'agent principal situé dans la région principale.



Les agents principaux et de répliques agissent comme une solution de récupération de données entre régions. Si l'agent principal situé dans la région principale échoue, vous pouvez promouvoir l'agent de répliques situé dans la région secondaire au rang d'agent principal en lançant une commutation ou un basculement. L'ancien agent principal devient alors l'agent de répliques, et l'ancien agent de répliques est promu agent principal. Pour obtenir des instructions sur la création d'un agent principal et d'un agent de répliques, consultez Création d'un courtier de réplication de données entre régions Amazon MQ.

> **Note**
>
> Disponible uniquement pour les agents actifs/en veille.
> Non disponible pour les files d'attente en miroir.

Création d'un courtier de réplication de données entre régions Amazon MQ

Grâce à la réplication de données entre régions (CRDR), vous pouvez basculer entre les agents de messages Amazon MQ pour ActiveMQ dans deux régions AWS, selon vos besoins. Vous pouvez désigner un agent existant comme agent principal et créer une réplique pour cet agent, ou créer ensemble un nouvel agent principal et un nouvel agent de répliques. Vous pouvez ensuite promouvoir l'agent de répliques au rôle d'agent principal à l'aide de l'opération

d'API `Promote` Amazon MQ. Pour plus d'informations sur les agents principaux et les agents de répliques, consultez Brokers principaux et répliques pour la réplication de données entre régions.

Les instructions suivantes décrivent comment créer et configurer un agent de répliques à l'aide de la console de gestion Amazon MQ.

Rubriques

Prérequis

Pour utiliser la fonctionnalité de réplication de données inter-régions, vous devez vérifier et respecter les prérequis suivants :

- Version : la fonctionnalité de réplication de données inter-régions est uniquement disponible pour Amazon MQ pour les agents ActiveMQ dans la version 5.17.6 et supérieures.
- Région : la réplication de données inter-régions est prise en charge dans les régions suivantes : USA Est (Ohio), USA Est (Virginie du Nord), USA Ouest (Oregon) et USA Ouest (Californie du Nord).
- Type d'instance : la réplication de données inter-régions n'est disponible que pour les instances d'agent d'une taille de `mq.m5.large` ou supérieure.
- Type de déploiement : la réplication de données inter-régions n'est disponible que pour les agents actifs/en veille avec un déploiement dans plusieurs zones de disponibilité.
- État de l'agent : vous ne pouvez créer une réplique d'agent que pour un agent principal ayant le statut `Running`.

Étape 1 (facultative) : Créer un nouvel agent principal

Créer un nouvel agent principal

1. Connectez-vous à la console Amazon MQ.
2. Sur la page Agents de la console Amazon MQ, choisissez Créer les agents.
3. Dans la page Select broker engine (Sélectionner le moteur de l'agent), choisissez Apache ActiveMQ.

4. Dans la page Select deployement and storage (Sélectionner le déploiement et le stockage), dans la section Deployement mode and storage type (Mode de déploiement et type de stockage), procédez comme suit :

   - Pour Mode de déploiement, choisissez Déploiement actif/en veille pour une haute disponibilité. Un déploiement actif/en veille pour une haute disponibilité est composé de deux agents répartis dans deux zones de disponibilité différentes, configurés en une paire redondante. Ces agents communiquent de manière synchrone avec votre application et avec Amazon EFS. Pour de plus amples informations, veuillez consulter Options de déploiement pour Amazon MQ pour les courtiers ActiveMQ.

5. Choisissez Suivant.

6. Sur la page Configure settings (Configurer les paramètres), dans la section Details (Détails), effectuez ce qui suit :

   a. Renseignez Broker name (Nom de l'agent).

   > **Important**
   >
   > N'ajoutez pas de données d'identification personnelle (PII) ou d'autres données confidentielles ou sensibles dans les noms d'agents. Les noms des courtiers sont accessibles à d'autres AWS services, notamment CloudWatch aux journaux. Les noms d'agents ne sont pas destinées à être utilisées pour des données privées ou sensibles.

   b. Cliquez sur Broker instance type (Type d'instance de l'agent) (par exemple, mq.m5.large). Pour de plus amples informations, veuillez consulter Broker instance types.

7. Dans la section ActiveMQ Web Console access (Accès à la console web ActiveMQ), renseignez Username (Nom d'utilisateur) et Password (Mot de passe). Les restrictions suivantes s'appliquent aux noms d'utilisateur et aux mots de passe des agents :

   - Votre nom d'utilisateur peut contenir uniquement des caractères alphanumériques, des tirets, des points, des traits de soulignement et des tildes (- . _ ~).
   - Votre mot de passe doit comporter 12 caractères minimum, dont au moins 4 caractères uniques, et ne doit pas contenir de virgules, de deux-points ou de signes égal (,:=).

> **Important**
>
> N'ajoutez pas de données d'identification personnelle (PII) ou d'autres données confidentielles ou sensibles dans les noms d'utilisateur des agents. Les noms d'utilisateur des courtiers sont accessibles à d'autres AWS services, notamment aux CloudWatch journaux. Les noms d'utilisateur des agents ne sont pas destinées à être utilisées pour des données privées ou sensibles.

La barre d'éclair verte en haut de la page confirme qu'Amazon MQ est en train de créer l'agent de répliques dans la région de restauration. Vous pouvez également voir le rôle CRDR et le statut RPO de vos agents. Pour désactiver les colonnes Rôle CRDR et État du RPO, choisissez l'icône en forme d'engrenage dans le coin supérieur droit de la table Agents. Ensuite, sur la page Préférences, désactivez Rôle CRDR ou État du RPO.

Étape 2 : Créer une réplique d'un agent existant

1. Sur la page Agents de la console Amazon MQ, choisissez Créer un agent de réplica.
2. Sur la page Choisir l'agent principal, sélectionnez un agent existant à utiliser comme agent principal de réplication CRDR. Ensuite, choisissez Suivant.
3. Sur la page Configurer l'agent de répliques, utilisez le menu déroulant pour choisir la région de répliques.
4. Dans la section Utilisateur de la console ActiveMQ pour l'agent de répliques, fournissez un Nom d'utilisateur et un Mot de passe pour l'utilisateur de la console d'agent de répliques. Les restrictions suivantes s'appliquent aux noms d'utilisateur et aux mots de passe des agents :

   - Votre nom d'utilisateur peut contenir uniquement des caractères alphanumériques, des tirets, des points, des traits de soulignement et des tildes (- . _ ~).
   - Votre mot de passe doit comporter 12 caractères minimum, dont au moins 4 caractères uniques, et ne doit pas contenir de virgules, de deux-points ou de signes égal (,:=).

> **Important**
>
> N'ajoutez pas de données d'identification personnelle (PII) ou d'autres données confidentielles ou sensibles dans les noms d'utilisateur des agents. Les noms

d'utilisateur des courtiers sont accessibles à d'autres AWS services, notamment aux CloudWatch journaux. Les noms d'utilisateur des agents ne sont pas destinées à être utilisées pour des données privées ou sensibles.

5. Dans la section Utilisateur de réplication des données pour établir un pont entre les agents, fournissez un Nom d'utilisateur et un Mot de passe pour l'utilisateur qui accèdera à la fois à l'agent principal et à l'agent de répliques. Les restrictions suivantes s'appliquent aux noms d'utilisateur et aux mots de passe des agents :

   - Votre nom d'utilisateur peut contenir uniquement des caractères alphanumériques, des tirets, des points, des traits de soulignement et des tildes (- . _ ~).
   - Votre mot de passe doit comporter 12 caractères minimum, dont au moins 4 caractères uniques, et ne doit pas contenir de virgules, de deux-points ou de signes égal (,:=).

   Important

   N'ajoutez pas de données d'identification personnelle (PII) ou d'autres données confidentielles ou sensibles dans les noms d'utilisateur des agents. Les noms d'utilisateur des courtiers sont accessibles à d'autres AWS services, notamment aux CloudWatch journaux. Les noms d'utilisateur des agents ne sont pas destinées à être utilisées pour des données privées ou sensibles.

   Configurez tous les paramètres supplémentaires. Ensuite, choisissez Suivant.

6. Sur la page Vérifier et créer, passez en revue les détails de l'agent de répliques, puis choisissez Créer un agent de réplica.
7. Ensuite, redémarrez l'agent principal. Cela redémarrera également l'agent de répliques. Pour obtenir des instructions sur le redémarrage de votre agent, consultez Rebooting a Broker.

Pour plus d'informations sur la configuration de paramètres supplémentaires pour votre agent ActiveMQ, consultez Mise en route : création et connexion à un courtier ActiveMQ.

Suppression d'un courtier de réplication de données entre régions Amazon MQ

Pour supprimer un courtier CRDR (Inter-Region Data Replication) principal ou répliqué, vous devez d'abord dissocier puis redémarrer les courtiers. Les instructions suivantes indiquent comment dissocier et redémarrer les courtiers à l'aide de la console AWS de gestion.

1. Sur la page Agents, sélectionnez l'agent CRDR que vous souhaitez dissocier, puis choisissez Modifier.
2. Sur la page Modifier de l'agent, dans la section Réplication des données, choisissez Dissocier les agents.
3. Entrez « confirmer » dans la fenêtre contextuelle pour confirmer votre choix. Choisissez ensuite Dissocier les agents.
4. Ensuite, redémarrez l'agent principal dissocié. Cela redémarrera également l'agent de répliques. Pour obtenir des instructions sur le redémarrage de votre agent, consultez Rebooting a Broker. Après le redémarrage de l'agent principal, les deux agents sont dissociés et peuvent être supprimés individuellement. Pour supprimer votre agent, consultez Deleting a broker.

Initier un basculement ou un basculement pour promouvoir le rôle de courtier principal d'Amazon MQ en tant que courtier principal

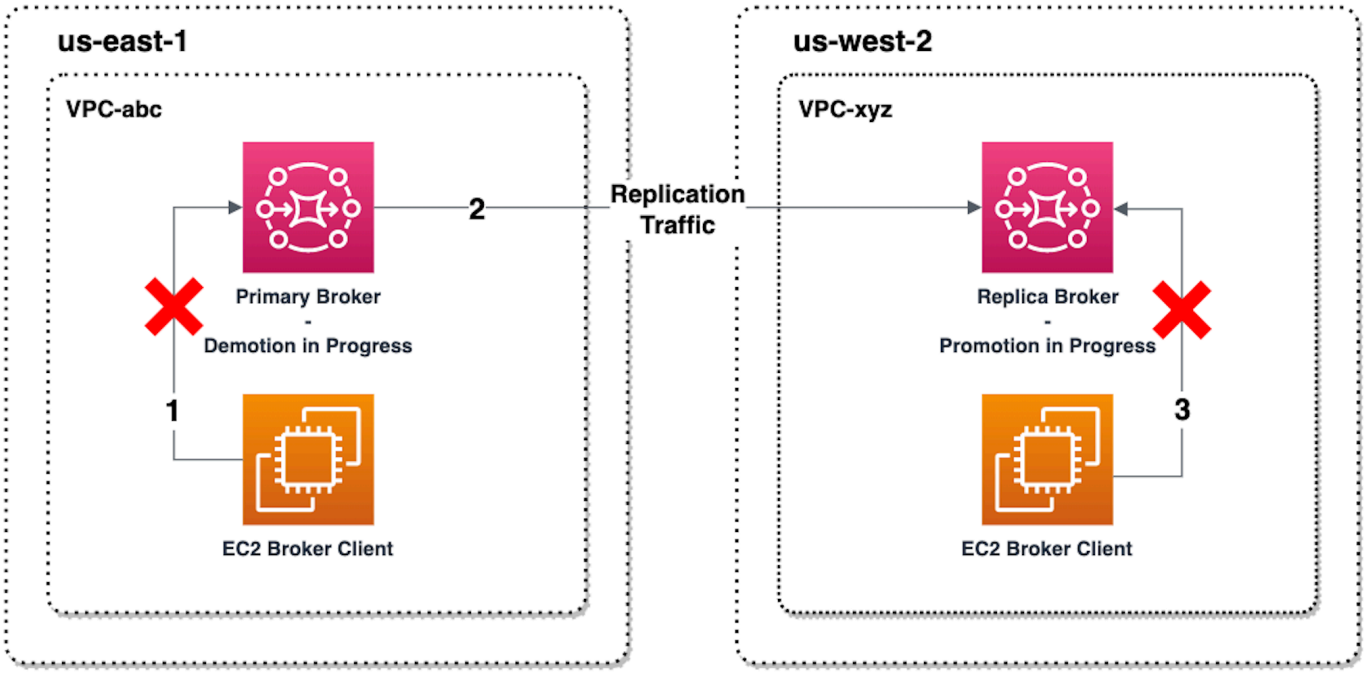
Vous pouvez déclencher une commutation ou un basculement lorsque vous voulez promouvoir l'agent de répliques au rôle d'agent principal. Lorsque vous promouvez l'agent de répliques, l'agent principal est rétrogradé au rôle d'agent de répliques.

Une commutation privilégie la cohérence par rapport à la disponibilité. Les agents ont la garantie d'avoir un état identique à la fin de cette opération de basculement. Lors d'une commutation, il peut y avoir une période pendant laquelle aucun agent n'est disponible pour les connexions client, alors que la cohérence entre agents est établie. Les deux agents auront le même état au moment de la promotion de la réplique. La réussite de la commutation dépend de la santé des deux régions et du réseau entre régions.

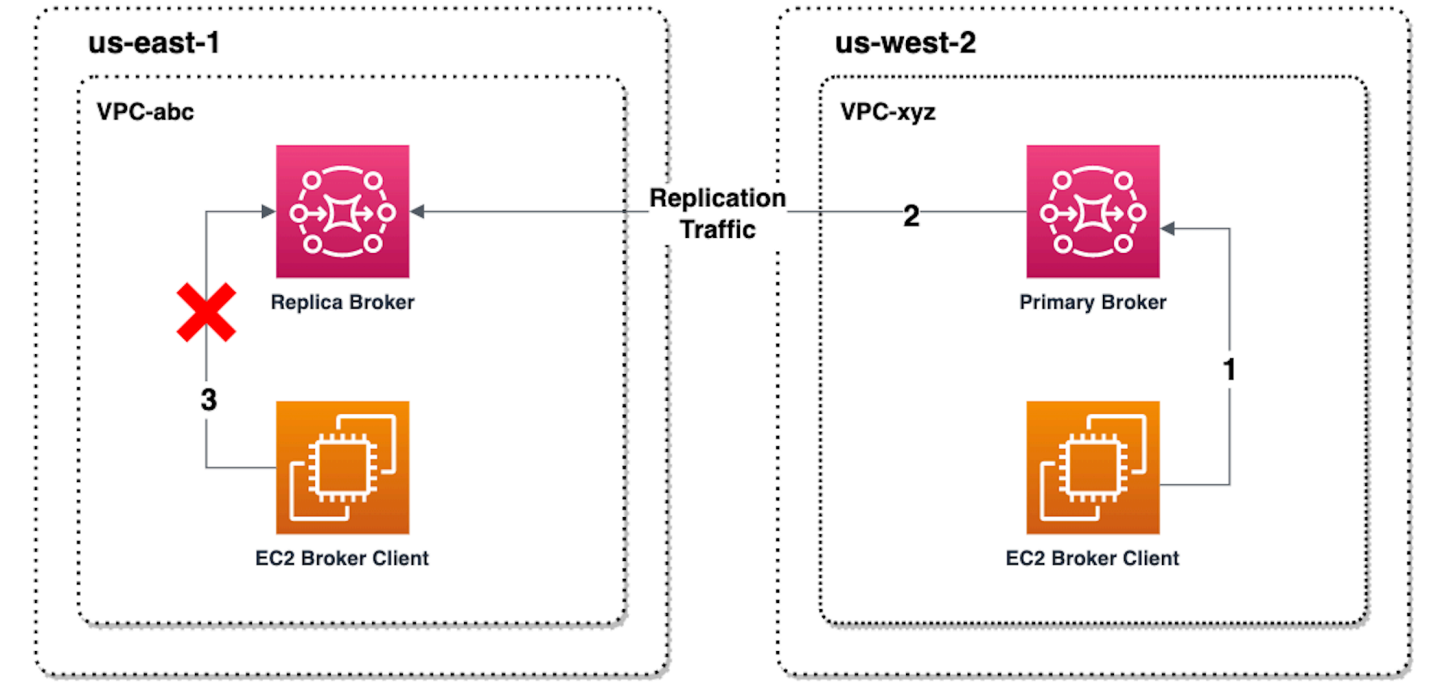
Un basculement privilégie la disponibilité par rapport à la cohérence. Il n'est pas garanti que les courtiers auront des états identiques à la fin de cette opération. Lors d'un basculement, l'agent de répliques est assuré de devenir immédiatement disponible pour traiter le trafic client, sans attendre que des données de réplication soient synchronisées ou que le serveur principal reçoive le signal

d'arrêt. La réussite du basculement ne dépend ni de la santé de la région principale d'origine ni du réseau entre régions.

Le schéma suivant illustre une commutation dans laquelle aucun agent n'accepte les connexions client alors que la file d'attente de réplication est vidée et que les états des agents sont synchronisés. Dans ce processus, le client du VPC du courtier principal n'est pas en mesure de produire d'autres changements d'état pendant que l'opération est en cours, et le courtier principal est rétrogradé au rang de réplique. Lorsque la file d'attente de réplication est vidée et que les deux agents atteignent un état identique, le client du VPC de l'agent de répliques ne peut pas se connecter à l'agent de répliques tant que l'opération de basculement n'est pas terminée, et l'agent de répliques est promu au rang d'agent principal.



Le schéma suivant illustre le statut des agents une fois le processus de commutation terminé. L'agent de répliques d'origine a désormais été promu au rôle d'agent principal et il accepte les connexions client. Le client peut produire et consommer les données provenant de l'agent.



Promotion de l'agent de répliques à l'aide de la console

Pour promouvoir l'agent de répliques par le biais de la commutation ou du basculement, procédez comme suit dans la console Amazon MQ.

> **Note**
>
> Vous ne pouvez pas déclencher la commutation ou le basculement sur un agent principal.

1. Passez à la région de votre agent de répliques. Sur votre table Agents, sélectionnez l'agent de répliques existant que vous allez promouvoir au rang d'agent principal.
2. Sur la page Détails de l'agent, procédez comme suit :
    1. Sélectionnez Promouvoir la réplique.
    2. Dans la fenêtre contextuelle, choisissez Commutation ou Basculement.
    3. Tapez « confirm » dans la zone de texte pour confirmer votre choix.
    4. Choisissez Confirmer.

Après avoir lancé le basculement, le statut de l'agent passe à Basculement en cours. La barre de progression bleue en haut de la page Agents devient verte lorsque le basculement est terminé.

> **Note**
>
> La configuration est uniquement répliquée au moment de la création de l'agent de répliques. Aucune mise à jour ultérieure n'est répliquée.

Mesures de réplication de données entre régions sur Amazon CloudWatch

La fonctionnalité de réplication de données entre régions d'Amazon MQ pour ActiveMQ fournit des métriques permettant de maintenir la fiabilité, la disponibilité et les performances de vos agents principaux et de répliques. Au cours du processus de réplication, un agent de répliques situé dans une région secondaire reçoit des données répliquées de manière asynchrone de la part de l'agent principal situé dans la région principale. Si l'agent principal situé dans la région principale échoue, vous pouvez promouvoir l'agent de répliques situé dans la région secondaire au rang d'agent principal en lançant une commutation ou un basculement. Pour obtenir des instructions sur l'affichage des statistiques sur Amazon CloudWatch, consultezAccès aux CloudWatch métriques pour Amazon MQ.

Horodatages de réplication CRDR

Les horodatages suivants décrivent le mode de calcul des métriques trouvées sur Amazon CloudWatch . Le processus de réplication des données comporte cinq horodatages :

- Heure d'observation actuelle (TCO) : l'instant présent dans le temps.
- Heure de création (TC) : l'instant où un événement a été créé dans la file d'attente de réplication par l'agent principal. Disponible à la fois pour les agents principaux et les agents de répliques.
- Heure de livraison (TD) : l'instant où un événement a été transmis avec succès à l'agent de répliques. Disponible uniquement sur les agents de répliques.
- Heure de traitement (TP) : l'instant où un événement a été traité avec succès par l'agent de répliques. Disponible uniquement sur les agents de répliques.
- Heure d'accusé de réception (TA) : l'instant où un événement a été reconnu avec succès par l'agent principal. Disponible uniquement sur les agents principaux.

Estimez les performances de basculement et de basculement à l'aide des métriques CRDR CloudWatch

Amazon MQ active les mesures pour votre agent par défaut. Vous pouvez consulter les statistiques de votre courtier en accédant à la CloudWatch console Amazon ou en utilisant l' CloudWatch API. Les métriques suivantes sont utiles pour comprendre les performances de réplication et de commutation/basculement de vos agents CRDR :

Métrique Amazon MQ CloudWatch	Raison de l'utilisation de la réplication CRDR
`TotalReplicationLag`	Temps estimé entre TA et TC du dernier événement non reconnu sur l'agent principal.
`ReplicationLag`	Temps estimé entre TP et TC du dernier événement non reconnu sur l'agent de répliques.
`PrimaryWaitTime`	Temps estimé entre TCO et TC du dernier événement traité sur l'agent principal.
`ReplicaWaitTime`	Temps estimé entre TCO et TP du dernier événement traité sur l'agent de répliques.
`QueueSize`	Nombre total d'événements non reconnus dans la file d'attente de réplication sur l'agent principal.

`TotalReplicationLag` et `ReplicationLag` décrivent la réplication différée entre l'agent principal et l'agent de répliques. Ces deux métriques peuvent également être utilisées pour estimer le temps jusqu'à la fin de l'opération de commutation ou de basculement en cours.

`PrimaryWaitTime` et `ReplicaWaitTime` peuvent être utilisés pour identifier les problèmes en cours liés au processus de réplication. Si la valeur de cette métrique augmente constamment, cela peut indiquer que le processus de réplication est dégradé ou suspendu. La réplication peut être lente en raison de problèmes tels que le partitionnement du réseau, le démarrage de l'agent et la lenteur de la restauration.

Didacticiels ActiveMQ

Les didacticiels suivants vous montrent comment créer et connecter vos agents ActiveMQ. Pour utiliser l'exemple de code Java ActiveMQ, vous devez installer le kit de développement Java édition Standard et apporter des modifications de code.

Rubriques

- Création et configuration d'un réseau d'agents Amazon MQ
- Connexion d'une application Java à votre agent Amazon MQ
- Intégration des agents ActiveMQ avec LDAP
- Étape 3 : (Facultatif) Se connecter à une AWS Lambda fonction
- Création d'un utilisateur de courtier ActiveMQ
- Modifier un utilisateur de courtier ActiveMQ
- Supprimer un utilisateur de courtier ActiveMQ
- Exemples pratiques d'utilisation de Java Message Service (JMS) avec ActiveMQ

Création et configuration d'un réseau d'agents Amazon MQ

Un réseau d'agents est composé de plusieurs agents à instance unique actifs simultanément ou plusieurs agents actifs/en veille. Dans ce didacticiel, vous allez apprendre à créer un réseau d'agents à deux agents avec une topologie source et puits.

Pour une présentation conceptuelle et des informations de configuration détaillées, consultez les sections suivantes :

- Réseau de courtiers Amazon MQ
- Correctement configurer votre réseau d'agents
- `networkConnector`
- `networkConnectionStartAsynchrone`

- Réseaux d'agents dans la documentation ActiveMQ

Vous pouvez utiliser la console Amazon MQ pour créer un réseau d'agents Amazon MQ. Puisque vous pouvez démarrer la création des deux agents en parallèle, ce processus dure environ 15 minutes.

Rubriques

Conditions préalables

Pour créer un réseau d'agents, vous devez disposer des éléments suivants :

- Deux ou plusieurs agents actifs simultanément (nommés `MyBroker1` et `MyBroker2` dans ce didacticiel). Pour plus d'informations sur la création d'agents, consultez le didacticiel Mise en route : création et connexion à un courtier ActiveMQ.
- Les deux courtiers doivent appartenir au même VPC ou être pairs. VPCs Pour plus d'informations VPCs, consultez Qu'est-ce qu'Amazon VPC ? dans le guide de l'utilisateur Amazon VPC et qu'est-ce que le peering VPC ? dans le guide de peering Amazon VPC.

  > **Important**
  >
  > Si vous n'avez pas de VPC par défaut, de sous-réseau(x) ou de groupe de sécurité, vous devez les créer en premier. Pour plus d'informations, consultez ce qui suit dans le Guide de l'utilisateur Amazon VPC :
  >
  > - Création d'un VPC par défaut
  > - Création d'un sous-réseau par défaut
  > - Création d'un groupe de sécurité

- Deux utilisateurs avec des informations d'identification de connexion identiques pour les deux agents. Pour plus d'informations sur la création d'utilisateurs, consultez Création d'un utilisateur de courtier ActiveMQ.

> **Note**
>
> Lors de l'intégration de l'authentification LDAP à un réseau d'agents, assurez-vous que l'utilisateur existe à la fois en tant qu'agents ActiveMQ et en tant qu'utilisateur LDAP.

L'exemple suivant utilise deux agents à instance unique. Cependant, vous pouvez créer des réseaux d'agents à l'aide d'agents actifs/en veille ou d'une combinaison des modes de déploiement d'agents.

Étape 1 : Autoriser le trafic entre les agents

Une fois que vous avez créé vos agents, vous devez autoriser le trafic entre eux.

1. Sur la console Amazon MQ, sur la page MyBroker2, dans la section Détails, sous Sécurité et réseau, choisissez le nom de votre groupe de sécurité ou.

   La page Groupes de sécurité du tableau de bord EC2 est affichée.
2. Dans la liste des groupes de sécurité, choisissez votre groupe de sécurité.
3. Au bas de la page, choisissez Entrant, puis Modifier.
4. Dans la boîte de dialogue Modifier les règles entrantes, ajoutez une règle pour le OpenWire point de terminaison.
   a. Choisissez Add Rule (Ajouter une règle).
   b. Pour Type, sélectionnez Custom TCP (TCP personnalisé).
   c. Pour Port Range, tapez le OpenWire port (`61617`).
   d. Effectuez l’une des actions suivantes :
      - Si vous souhaitez limiter l'accès à une adresse IP en particulier, pour Source, laissez Personnalisé sélectionné, puis saisissez l'adresse IP de `MyBroker1`, suivie de `/32`. (Cela convertit l'adresse IP en un enregistrement CIDR valide). Pour plus d'informations, consultez Interfaces réseau Elastic.

> **Tip**
>
> Pour extraire l'adresse IP de `MyBroker1`, sur la console Amazon MQ, choisissez le nom de l'agent et accédez à la section Details (Détails).

- Si tous vos agents sont privés et appartiennent au même VPC, pour Source, laissez Personnalisé sélectionné, puis saisissez l'ID du groupe de sécurité que vous modifiez.

> **Note**
>
> Pour les agents publics, vous devez limiter l'accès à l'aide d'adresses IP.

e. Choisissez Enregistrer.

Votre agent peut désormais accepter les connexions entrantes.

Étape 2 : Configurer le réseau de connecteurs pour votre agent

Une fois le trafic autorisé entre vos agents, vous devez configurer les connecteurs de réseau pour l'un d'entre eux.

1. Modifiez la révision de configuration pour l'agent `MyBroker1`.

   a. Sur la page MyBroker1, choisissez Modifier.

   b. Sur la page Modifier MyBroker 1, dans la section Configuration, choisissez Afficher.

   Le type de moteur de l'agent et la version que la configuration utilise (par exemple, Apache ActiveMQ 5.15.0) sont affichés.

   c. Dans l'onglet Configuration details, le numéro de révision de configuration, la description et la configuration d'agent au format XML sont affichés.

   d. Choisissez Modifier la configuration.

   e. En bas du fichier de configuration, supprimez la section `<networkConnectors>` et incluez les informations suivantes :

   - Le `name` du connecteur de réseau.
   - Le `username` de la console web ActiveMQ qui est commune aux deux agents.
   - Activer les connexions `duplex`.

- Effectuez l'une des actions suivantes :
  - Si vous connectez le broker à un broker à instance unique, utilisez le `static:` préfixe et le OpenWire point de terminaison `uri` pour. `MyBroker2` Par exemple :

```
<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>
```

  - Si vous connectez le courtier à un courtier actif/de secours, utilisez le `static+failover` transport et le point de OpenWire terminaison `uri` pour les deux courtiers avec les paramètres de requête suivants. `?randomize=false&maxReconnectAttempts=0` Par exemple :

```
<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**
>
> N'incluez pas les informations d'identification de connexion pour l'utilisateur ActiveMQ.

f. Choisissez Enregistrer.

g. Dans la boîte de dialogue Save revision (Enregistrer la révision), tapez `Add network of brokers connector for MyBroker2`.

h. Choisissez Enregistrer pour enregistrer la nouvelle révision de la configuration.

2. Modifier l'agent `MyBroker1` pour définir la dernière révision de configuration comme s'appliquant immédiatement.

   a. Sur la page MyBroker1, choisissez Modifier.

b. Sur la page Modifier MyBroker 1, dans la section Configuration, sélectionnez Planifier les modifications.

c. Dans la section Schedule broker modifications (Planifier les modifications de l'agent), choisissez d'appliquer les modifications immédiatement.

d. Cliquez sur Appliquer.

L'agent `MyBroker1` est redémarré et votre révision de configuration est appliquée.

Le réseau d'agents est créé.

Étapes suivantes

Une fois votre réseau d'agents configuré, vous pouvez le tester en produisant et en consommant des messages.

> **Important**
>
> Assurez-vous d'activer les connexions entrantes depuis votre machine locale pour le broker `MyBroker1` sur le port 8162 (pour la console Web ActiveMQ) et le port 61617 (pour le point de terminaison). OpenWire
> Il se peut également que vous ayez besoin de régler les paramètres de votre(vos) groupe(s) de sécurité afin d'autoriser le producteur et le consommateur à se connecter au réseau d'agents.

1. Sur la console Amazon MQ, accédez à la section Connections (Connexions) et notez le point de terminaison de la console web ActiveMQ pour l'agent `MyBroker1`.
2. Accédez à la console web ActiveMQ pour l'agent `MyBroker1`.
3. Pour vérifier que le pont réseau est connecté, choisissez Réseau.

   Dans la section Network Bridges (Ponts de réseau), le nom et l'adresse du `MyBroker2` sont listés dans les colonnes Remote Broker (Agent à distance) et Remote Address (Adresse à distance).
4. À partir de n'importe quelle machine ayant accès à l'agent `MyBroker2`, créez un consommateur. Par exemple :

```
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
```

Le consommateur se connecte au OpenWire point de terminaison de `MyBroker2` et commence à consommer les messages de la file d'attente`MyQueue`.

5. À partir de n'importe quelle machine ayant accès à l'agent `MyBroker1`, créez un producteur et envoyez quelques messages. Par exemple :

```
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
```

Le producteur se connecte au OpenWire point de terminaison de `MyBroker1` et commence à produire des messages persistants à mettre en file d'attente`MyQueue`.

Connexion d'une application Java à votre agent Amazon MQ

Après avoir créé un agent ActiveMQ Amazon MQ, vous pouvez y connecter votre application. Les exemples suivants montrent comment utiliser Java Message Service (JMS) pour créer une connexion à l'agent, créer une file d'attente et envoyer un message. Pour un exemple Java complet et fonctionnel, consultez Working Java Example.

Vous pouvez vous connecter à des agents ActiveMQ à l'aide de différents clients ActiveMQ. Nous vous recommandons d'utiliser le client ActiveMQ.

Rubriques

Conditions préalables

Activer les attributs du VPC

Pour vous assurer que votre agent est accessible dans votre VPC, vous devez activer les attributs `enableDnsHostnames` et `enableDnsSupport` du VPC. Pour plus d'informations, consultez Prise en charge du DNS dans votre VPC dans le Guide de l'utilisateur Amazon VPC.

Activation des connexions entrantes

Activez ensuite les connexions entrantes pour votre application.

1. Connectez-vous à la console Amazon MQ.
2. Dans la liste des courtiers, choisissez le nom de votre courtier (par exemple, MyBroker).
3. Sur la ***`MyBroker`***page, dans la section Connexions, notez les adresses et les ports de l'URL de la console Web du courtier et des protocoles au niveau du fil.
4. Dans la section Details (Détails), sous Security and network (Sécurité et réseau), choisissez le nom de votre groupe de sécurité ou

   La page Groupes de sécurité du tableau de bord EC2 est affichée.
5. Dans la liste des groupes de sécurité, choisissez votre groupe de sécurité.
6. Au bas de la page, choisissez Entrant, puis Modifier.
7. Dans la boîte de dialogue Edit inbound rules (Modifier les règles entrantes), ajoutez une règle pour chaque URL ou point de terminaison pour qu'ils soient accessibles publiquement (l'exemple suivant montre comment procéder pour une console web d'agent).

   a. Choisissez Add Rule (Ajouter une règle).

   b. Pour Type, sélectionnez Custom TCP (TCP personnalisé).

   c. Pour Port Range (Plage de ports), saisissez le port de la console web (`8162`).

   d. Pour Source, laissez l'option Custom (Personnalisée) sélectionnée, puis tapez l'adresse IP du système qui doit pouvoir accéder à la console web (par exemple, `192.0.2.1`).

   e. Choisissez Enregistrer.

   Votre agent peut désormais accepter les connexions entrantes.

Ajout de dépendances Java

Ajoutez les packages `activemq-client.jar` et `activemq-pool.jar` au chemin de classe Java. L'exemple suivant illustre ces dépendances dans un fichier `pom.xml` de projet 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>
```

Pour plus d'informations sur `activemq-client.jar`, consultez Configuration initiale dans la documentation ActiveMQ Apache.

> **Important**
>
> Dans l'exemple de code suivant, les producteurs et les consommateurs s'exécutent dans un seul thread. Pour les systèmes de production (ou pour tester le basculement d'instance d'agent), assurez-vous que vos producteurs et vos consommateurs s'exécutent sur des hôtes ou des threads distincts.

Pour créer un producteur de messages et envoyer un message

Suivez les instructions ci-dessous pour créer un générateur de message et recevoir un message.

1. Créez une fabrique de connexions groupées JMS pour le producteur de messages à l'aide du point de terminaison de votre agent, puis appelez la méthode `createConnection` par rapport à la fabrique.

   > **Note**
   >
   > Pour un active/standby courtier, Amazon MQ fournit deux consoles Web ActiveMQ URLs, mais une seule URL est active à la fois. De même, Amazon MQ fournit deux

points de terminaison pour chaque protocole de niveau filaire, mais un seul point de terminaison est actif dans chaque paire à la fois. Les suffixes `-1` et `-2` indiquent une paire redondante. Pour de plus amples informations, veuillez consulter Options de déploiement pour Amazon MQ pour les courtiers ActiveMQ).
Pour les points de terminaison du protocole filaire, vous devez autoriser votre application à se connecter à l'un ou l'autre point de terminaison à l'aide du transport Failover.

```
// 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**
>
> Les producteurs de messages doivent toujours utiliser la classe `PooledConnectionFactory`. Pour de plus amples informations, veuillez consulter Toujours utiliser le regroupement de connexions.

2. Créez une session, une file d'attente nommée `MyQueue` et un producteur de messages.

```
// 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. Créez la chaîne de message `"Hello from Amazon MQ!"`, puis envoyez le message.

```
// 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. Nettoyez le producteur.

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

Pour créer un consommateur de messages et recevoir le message

Suivez les instructions ci-dessous pour créer un générateur de message et recevoir un message.

1. Créez une fabrique de connexions JMS pour le producteur de messages à l'aide du point de terminaison de votre agent, puis appelez la méthode `createConnection` par rapport à la fabrique.

```
// 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**
>
> Les consommateurs de messages ne doivent jamais utiliser la classe `PooledConnectionFactory`. Pour de plus amples informations, veuillez consulter Toujours utiliser le regroupement de connexions.

2. Créez une session, une file d'attente nommée `MyQueue` et un consommateur de messages.

```
// 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. Commencez à attendre les messages et recevez le message lorsqu'il arrive.

```
// 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**
>
> Contrairement aux services de AWS messagerie (tels qu'Amazon SQS), le consommateur est constamment connecté au courtier.

4. Fermez le consommateur, la session et la connexion.

```
consumer.close();
```

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

Intégration des agents ActiveMQ avec LDAP

> **Important**
>
> Amazon MQ ne prend pas en charge les certificats de serveur émis par une autorité de certification privée.

Vous pouvez accéder à vos agents ActiveMQ en utilisant les protocoles suivants avec TLS activé :

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

Amazon MQ offre un choix entre l'authentification ActiveMQ native et l'authentification LDAP et l'autorisation pour gérer les autorisations utilisateur. Pour plus d'informations sur les restrictions liées aux noms d'utilisateur et aux mots de passe ActiveMQ, consultez Users.

Pour autoriser des utilisateurs et des groupes ActiveMQ à utiliser des files d'attente et des rubriques, vous devez modifier la configuration de votre agent. Amazon MQ utilise le plugin Simple Authentication d'ActiveMQ pour limiter la lecture et l'écriture aux destinations. Pour plus d'informations et d'exemples, consultez Toujours configurer un plan d'autorisation et `authorizationEntry`.

> **Note**
>
> Actuellement, Amazon MQ ne prend pas en charge l'authentification par certificat client.

Rubriques

Intégrer LDAP avec ActiveMQ

Vous pouvez authentifier les utilisateurs Amazon MQ à l'aide des informations d'identification stockées dans votre Active Directory ou un autre serveur LDAP. Vous pouvez également ajouter, supprimer et modifier des utilisateurs Amazon MQ et attribuer des autorisations aux rubriques et aux files d'attente. Les opérations de gestion telles que la création, la mise à jour et la suppression des agents nécessitent toujours des informations d'identification IAM et ne sont pas intégrées à LDAP.

Les clients qui souhaitent simplifier et centraliser leur authentification et leur autorisation d'agent Amazon MQ à l'aide d'un serveur LDAP peuvent utiliser cette fonctionnalité. La conservation de toutes les informations d'identification utilisateur sur le serveur LDAP permet d'économiser du temps et des efforts en fournissant un emplacement central pour stocker et gérer ces informations d'identification.

Amazon MQ fournit la prise en charge LDAP à l'aide du plugin Apache ActiveMQ JAAS. Tout serveur LDAP, tel que Microsoft Active Directory ou OpenLDAP pris en charge par le plugin, est également pris en charge par Amazon MQ. Pour de plus amples informations sur le plugin, veuillez consulter la section Sécurité de la documentation ActiveMQ.

Outre les utilisateurs, vous pouvez spécifier l'accès aux rubriques et aux files d'attente pour un groupe spécifique ou un utilisateur via votre serveur LDAP. Pour ce faire, créez des entrées représentant des rubriques et des files d'attente dans votre serveur LDAP, puis attribuez des autorisations à un utilisateur LDAP spécifique ou à un groupe. Vous pouvez ensuite configurer l'agent pour récupérer les données d'autorisation à partir du serveur LDAP.

> **Important**
>
> Lorsque vous utilisez LDAP, l'authentification ne distingue pas les majuscules des minuscules, mais l'autorisation fait la distinction entre majuscules et minuscules pour votre nom d'utilisateur.

Conditions préalables

Avant d'ajouter la prise en charge LDAP à un agent Amazon MQ nouveau ou existant, vous devez configurer un compte de service. Ce compte de service est requis pour initier une connexion à un serveur LDAP et doit disposer des autorisations appropriées pour établir cette connexion. Ce compte de service configurera l'authentification LDAP pour votre agent. Toutes les connexions client successives seront authentifiées via la même connexion.

Un compte de service est un compte de votre serveur LDAP qui a accès afin d'initier une connexion. Il s'agit d'une exigence LDAP standard et vous ne devez fournir les informations d'identification du compte de service qu'une seule fois. Une fois la connexion configurée, toutes les futures connexions client sont authentifiées via votre serveur LDAP. Les informations d'identification de votre compte de service sont stockées de manière sécurisée sous une forme chiffrée, accessible uniquement à Amazon MQ.

Pour intégrer ActiveMQ, une arborescence d'informations d'annuaire (DIT) spécifique est requise sur le serveur LDAP. Pour un exemple de fichier `ldif` qui montre clairement cette structure, consultez Import the following LDIF file into the LDAP server (Importer le fichier LDIF suivant dans le serveur LDAP) dans la section Security (Sécurité) de la documentation ActiveMQ.

Mise en route avec LDAP

Pour commencer, accédez à la console Amazon MQ et choisissez LDAP authentication and authorization (Authentification et autorisation LDAP) lorsque vous créez une instance d'agent existante ou nouvelle Amazon MQ.

Fournissez les informations suivantes sur le compte de service :

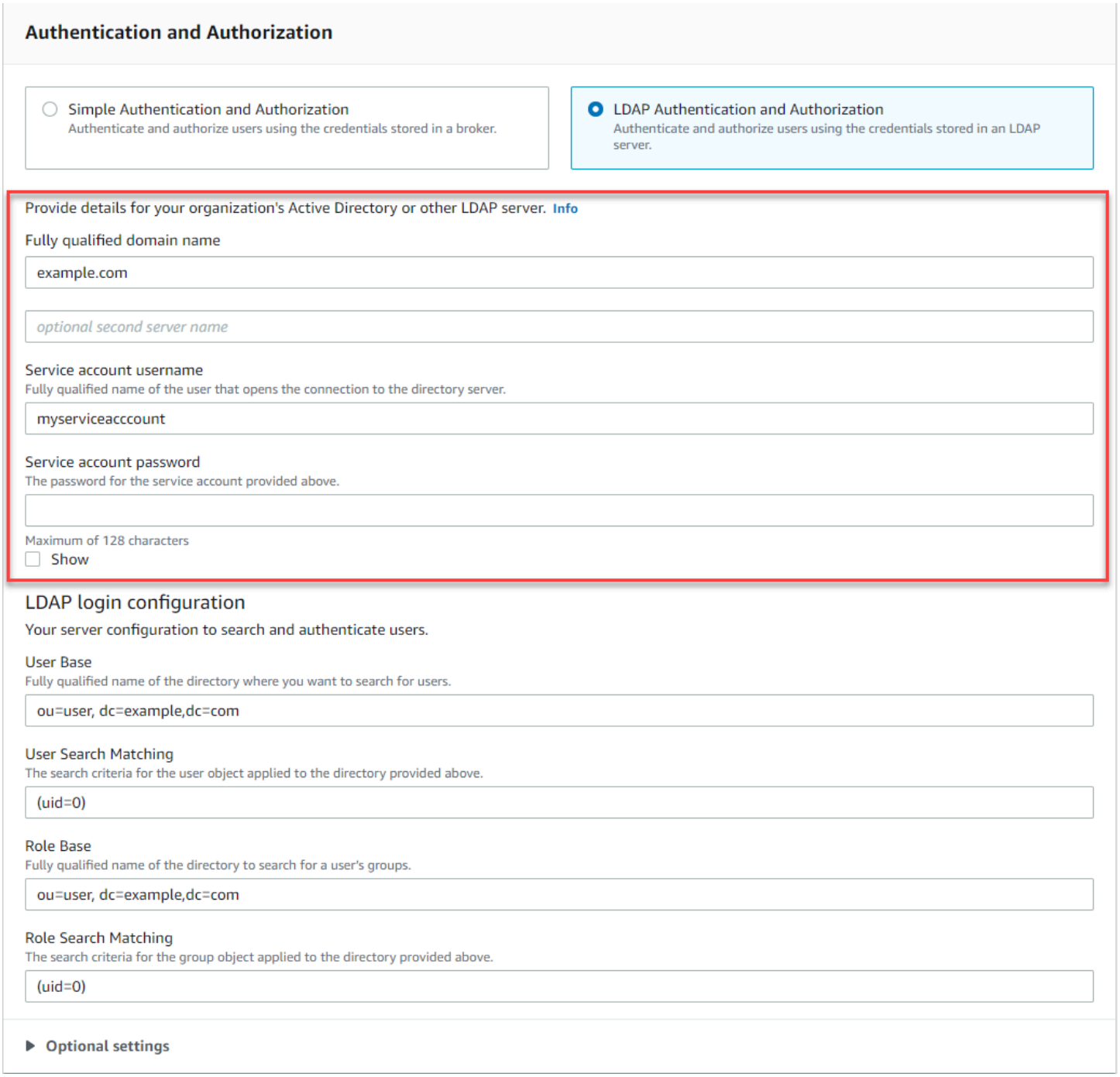
- Fully qualified domain name (Nom de domaine entièrement qualifié) : Emplacement du serveur LDAP auquel les demandes d'authentification et d'autorisation doivent être émises.

  > **Note**
  >
  > Le nom de domaine complet du serveur LDAP que vous fournissez ne doit pas inclure le numéro de protocole ou de port. Amazon MQ va ajouter le nom de domaine complet au protocole `ldaps`, et ajoutera le numéro de port `636`.
  > Par exemple, si vous fournissez le domaine complet suivant `example.com`, Amazon MQ accède à votre serveur LDAP à l'aide de l'URL suivante : `ldaps://example.com:636`.
  > Pour que l'hôte de l'agent puisse communiquer avec le serveur LDAP, le nom de domaine complet doit pouvoir être résolu publiquement. Pour garder le serveur LDAP privé et

sécurisé, limitez le trafic entrant dans les règles entrantes du serveur afin d'autoriser uniquement le trafic provenant du VPC de l'agent.

- Service account username (Nom d'utilisateur du compte de service) Nom unique de l'utilisateur qui sera utilisé pour effectuer la liaison initiale au serveur LDAP.
- Service account password (Mot de passe du compte de service) Mot de passe de l'utilisateur effectuant la liaison initiale.

L'image suivante met en évidence où fournir ces détails.



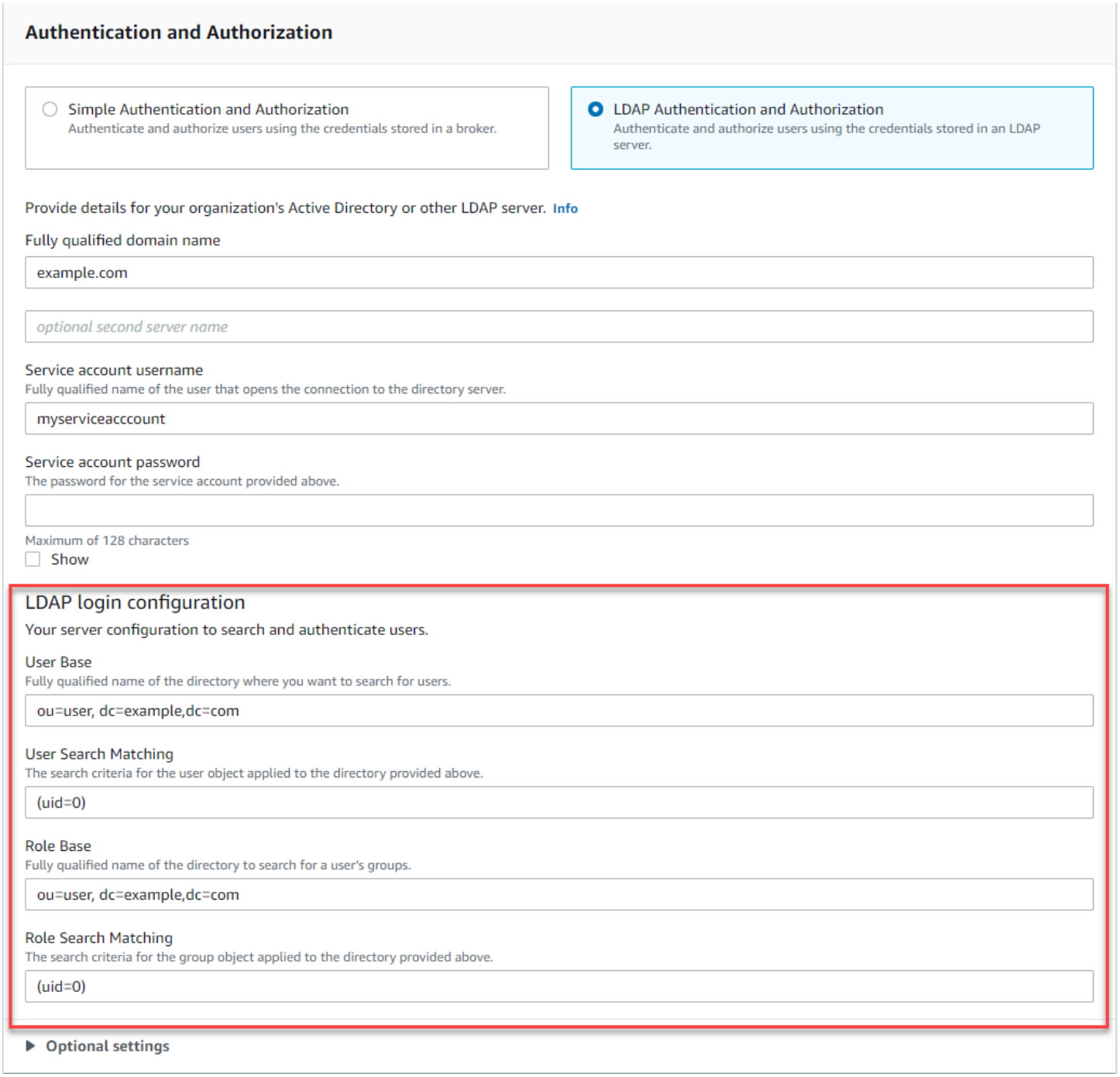
Dans la section LDAP login configuration (Configuration de la connexion LDAP), fournissez les informations requises suivantes :

- User Base (Base d'utilisateurs) Nom unique du nœud de l'arborescence des informations de répertoire (DIT) qui sera recherché pour les utilisateurs.
- User Search Matching (Recherche d'utilisateur) Filtre de recherche LDAP qui sera utilisé pour rechercher des utilisateurs dans `userBase`. Le nom d'utilisateur du client est remplacé

par l'espace réservé `{0}` dans le filtre de recherche. Pour plus d'informations, consultez Authentification et Autorisation.

- Role Base (Base de rôles) Nom unique du nœud du DIT qui sera recherché pour des rôles. Les rôles peuvent être configurés en tant qu'entrées de groupe LDAP explicites dans votre répertoire. Une entrée de rôle typique peut consister en un attribut pour le nom du rôle, tel que Nom commun, et un autre attribut, tel que `member`, avec des valeurs représentant les noms distinctifs ou les noms d'utilisateur des utilisateurs appartenant au groupe de rôles. Par exemple, compte tenu de l'unité administrative, `group`, vous pouvez fournir le nom distinctif suivant : `ou=group,dc=example,dc=com`.
- Role Search Matching (Recherche de rôle) Filtre de recherche LDAP qui sera utilisé pour rechercher des rôles dans `roleBase`. Le nom unique de l'utilisateur correspondant à `userSearchMatching` est remplacé dans l'espace réservé `{0}` du filtre de recherche. Le nom d'utilisateur du client sera remplacé par l'espace réservé `{1}`. Par exemple, si les entrées de rôle dans votre répertoire incluent un attribut nommé `member`, contenant les noms d'utilisateur de tous les utilisateurs de ce rôle, vous pouvez fournir le filtre de recherche suivant : `(member:=uid={1})`.

L'image suivante met en surbrillance l'endroit où spécifier ces détails.



Dans Optional settings (Paramètres facultatifs), vous pouvez fournir les informations facultatives suivantes :

- User Role Name (Nom du rôle utilisateur) Le nom de l'attribut LDAP dans l'entrée de répertoire de l'utilisateur aux fins de l'adhésion au groupe de l'utilisateur. Dans certains cas, les rôles utilisateur peuvent être identifiés par la valeur d'un attribut dans l'entrée de répertoire de l'utilisateur. L'option

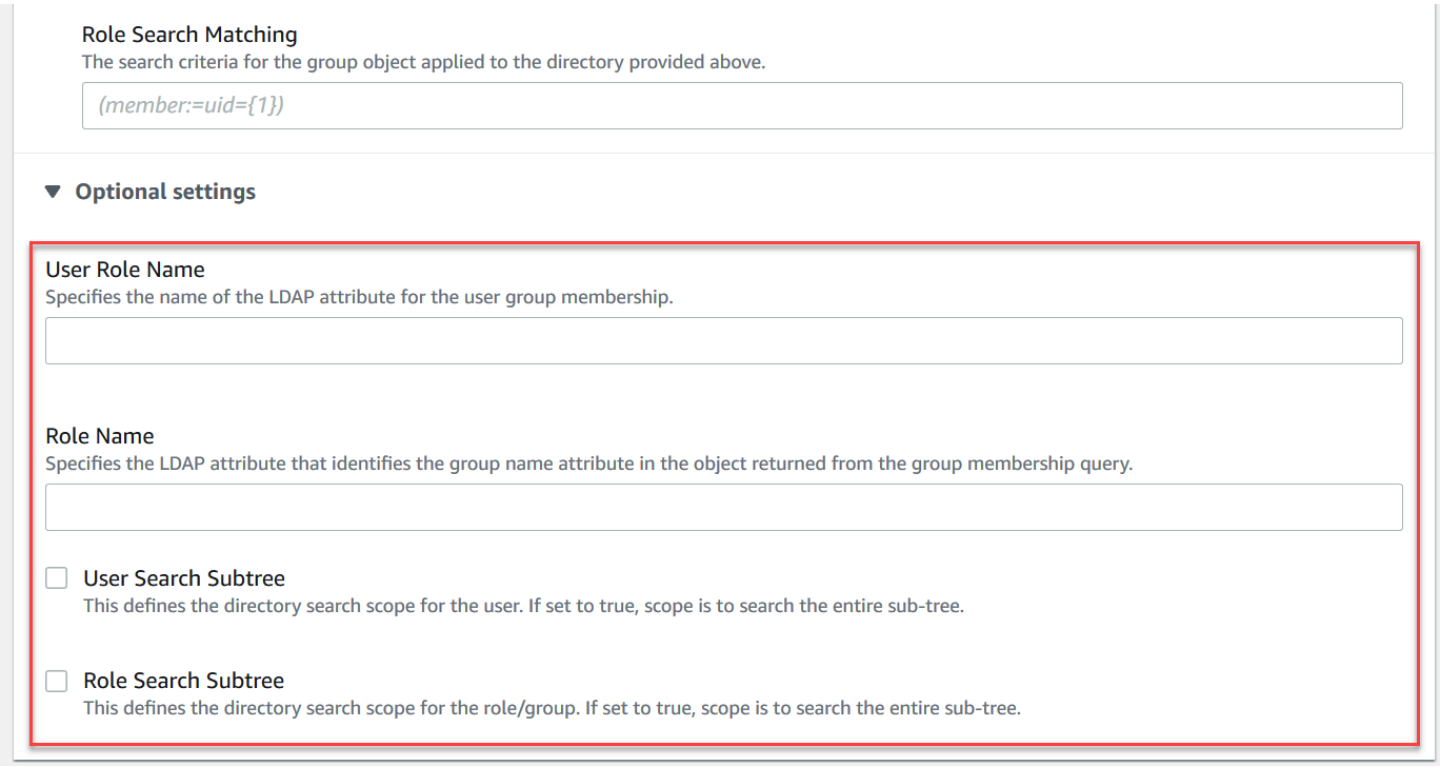
`userRoleName` vous permet de fournir le nom de cet attribut. Par exemple, considérons l'entrée utilisateur suivante :

```
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
```

Pour fournir le bon `userRoleName` pour l'exemple ci-dessus, spécifiez l'attribut `memberOf`. Si l'authentification réussit, le rôle `role1` est affecté à l'utilisateur.

- Role Name (Nom du rôle) L'attribut du nom de groupe dans une entrée de rôle dont la valeur constitue le nom de ce rôle. Par exemple, vous pouvez spécifier `cn` pour le nom commun d'une entrée de groupe. Si l'authentification réussit, l'utilisateur reçoit la valeur de l'attribut `cn` pour chaque entrée de rôle dont il est membre.
- User Search Subtree (Sous-arborescence de recherche d'utilisateur) Définit l'étendue de la requête de recherche utilisateur LDAP. Si true, la portée est définie pour rechercher la sous-arborescence entière sous le nœud défini par `userBase`.
- Role Search Subtree (Sous-arborescence de recherche de rôle) Définit l'étendue de la requête de recherche utilisateur LDAP. Si true, la portée est définie pour rechercher la sous-arborescence entière sous le nœud défini par `roleBase`.

L'image suivante met en surbrillance l'endroit où spécifier ces paramètres facultatifs.



Fonctionnement de l'intégration avec LDAP

Vous pouvez penser à l'intégration dans deux catégories principales : la structure pour l'authentification et la structure pour l'autorisation.

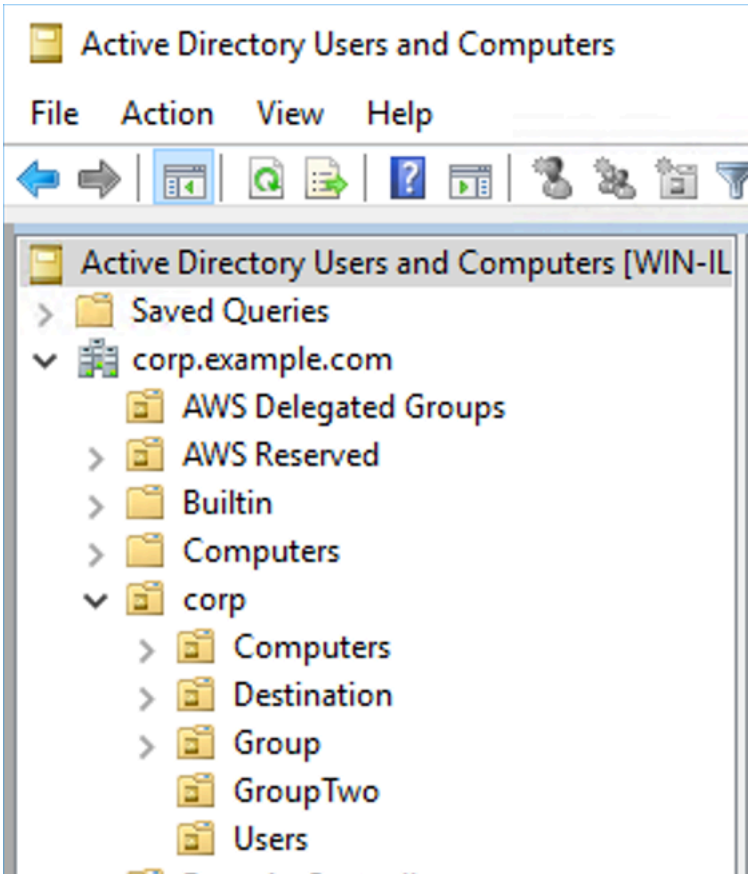
Authentification

Pour l'authentification, les informations d'identification du client doivent être valides. Ces informations d'identification sont validées par rapport aux utilisateurs de la base d'utilisateurs du serveur LDAP.

La base d'utilisateurs fournie à l'agent ActiveMQ doit pointer vers le nœud dans le DIT où les utilisateurs sont stockés sur le serveur LDAP. Par exemple, si vous utilisez AWS Managed Microsoft AD, et que vous avez les composants du domaine `corpexample`, et que vous avez au sein de ceux-`cicom`, vous avez des unités organisationnelles `corp` et `Users` que vous utiliseriez les éléments suivants comme base d'utilisateurs :

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

L'agent ActiveMQ recherche à cet emplacement dans le DIT les utilisateurs afin d'authentifier les demandes de connexion client auprès de l'agent.



Parce que le code source ActiveMQ code en dur le nom de l'attribut pour les utilisateurs sur `uid`, vous devez vous assurer que cet attribut est défini à chaque utilisateur. Pour plus de simplicité, vous pouvez utiliser le nom d'utilisateur de connexion de l'utilisateur. Pour plus d'informations, consultez le code source activemq et Configuration des mappages d'ID dans les utilisateurs et ordinateurs Active Directory pour les versions Windows Server 2016 (et ultérieures).

Pour activer l'accès à la console ActiveMQ pour des utilisateurs spécifiques, assurez-vous qu'ils appartiennent au `amazonmq-console-admins`.

Autorisation

Pour l'autorisation, les bases de recherche d'autorisations sont spécifiées dans la configuration de l'agent. L'autorisation est effectuée sur une base par destination (ou caractère générique, ensemble de destination) via l'élément `cachedLdapAuthorizationMap`, qui se trouve dans le fichier de configuration `activemq.xml` de l'agent. Pour de plus amples informations, consultez Module d'autorisation LDAP mis en cache.

> **Note**
>
> Pour pouvoir utiliser l'`cachedLDAPAuthorizationMap`élément dans le fichier de `activemq.xml` configuration de votre courtier, vous devez choisir l'option Authentification et autorisation LDAP lors de la création d'une configuration via le AWS Management Console, ou définir la création d'une configuration via le AWS Management Console, ou définir la `authenticationStrategy`propriété sur LDAP lors de la création d'une nouvelle configuration à l'aide de l'API Amazon MQ.

Vous devez fournir les trois attributs suivants dans l'élément `cachedLDAPAuthorizationMap` :

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

> **Important**
>
> Pour éviter que des informations sensibles ne soient directement placées dans le fichier de configuration de l'agent, Amazon MQ bloque l'utilisation des attributs suivants dans `cachedLdapAuthorizationMap` :
>
> - `connectionURL`
> - `connectionUsername`
> - `connectionPassword`
>
> Lorsque vous créez un courtier, Amazon MQ remplace les valeurs que vous fournissez par le biais ou dans la AWS Management Console`ldapServerMetadata`propriété de votre demande d'API par les attributs ci-dessus.

L'exemple suivant illustre un exemple d'utilisation de `cachedLdapAuthorizationMap`.

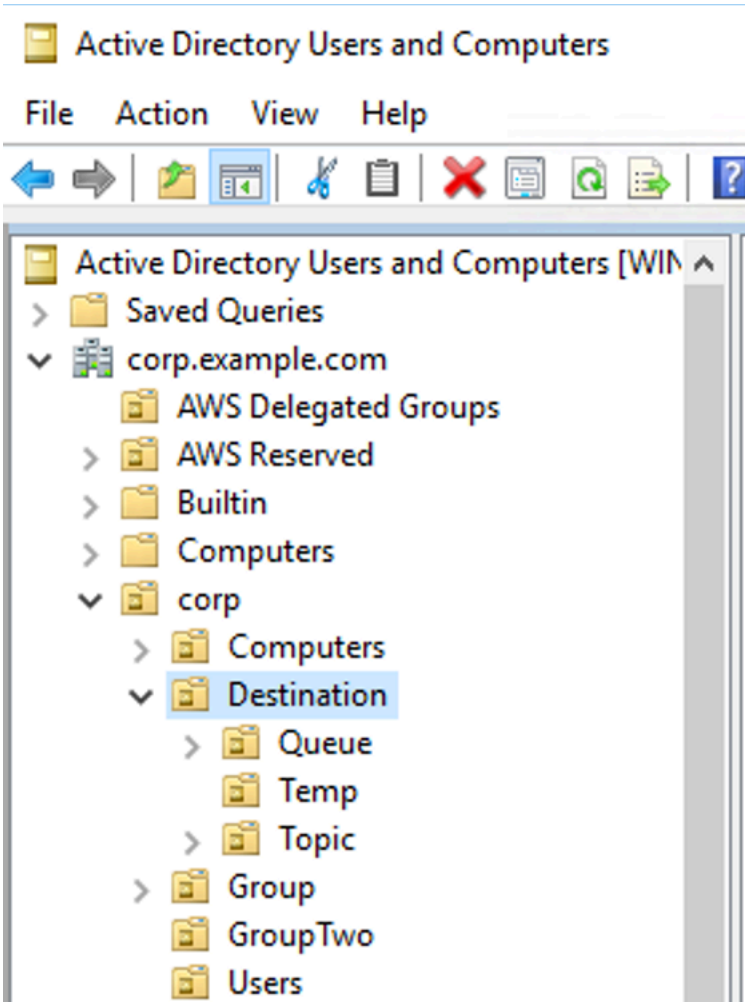
```
<authorizationPlugin>
    <map>
        <cachedLDAPAuthorizationMap
```

```
            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>
```

Ces valeurs identifient les emplacements dans le DIT où les autorisations pour chaque type de destination sont spécifiées. Ainsi, dans l'exemple ci-dessus AWS Managed Microsoft AD, en utilisant les mêmes composants de domaine que `corpexample`, et`com`, vous devez spécifier une unité organisationnelle nommée `destination` pour contenir tous vos types de destination. Dans cette unité d'organisation, vous en créeriez un pour `queues`, un pour `topics` et un pour destinations `temp`.

Cela signifie que votre base de recherche de file d'attente, qui fournit des informations d'autorisation pour les destinations de type file d'attente, aurait l'emplacement suivant dans votre DIT :

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



De même, les règles d'autorisation pour les rubriques et les destinations temporaires seraient situées au même niveau dans le 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
```

Dans l'unité d'organisation pour chaque type de destination (file d'attente, rubrique, temp), un caractère générique ou un nom de destination spécifique peut être fourni. Par exemple, pour fournir une règle d'autorisation pour toutes les files d'attente commençant par le préfixe DEMO.EVENTS.$., vous pouvez créer l'unité d'organisation suivante :

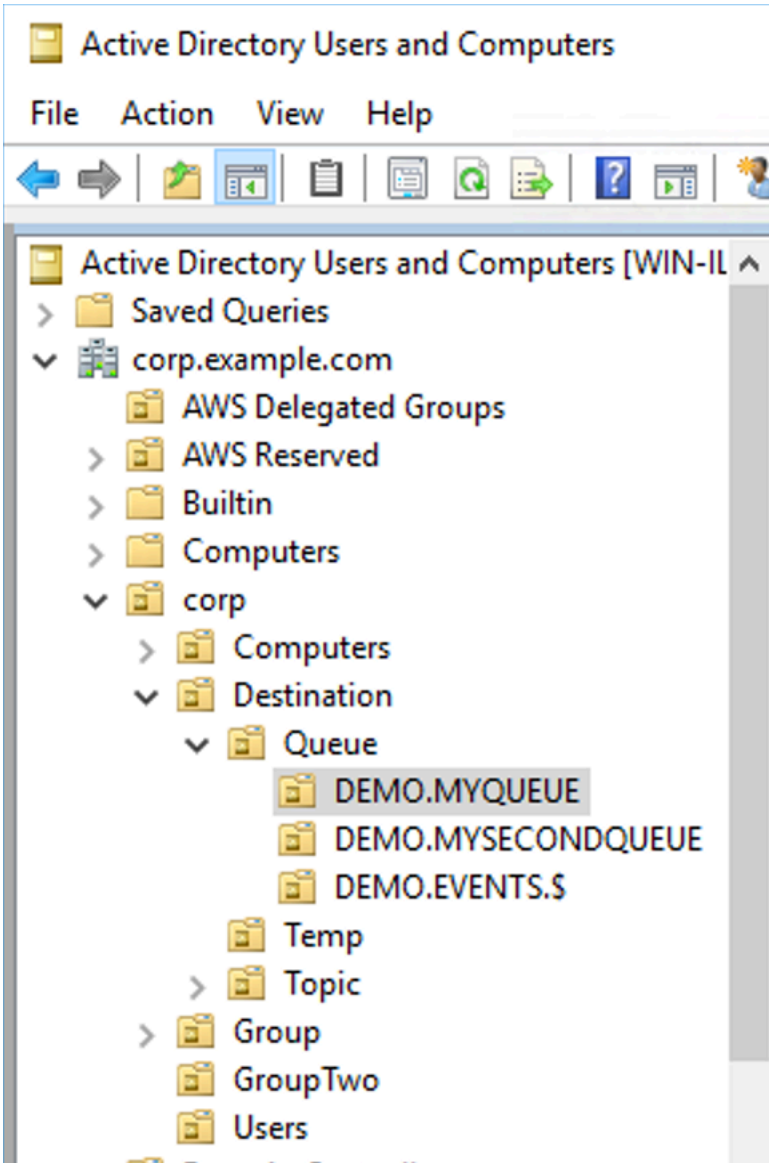
```
OU=DEMO.EVENTS.$,OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```

> **Note**
>
> L'unité d'organisation `DEMO.EVENTS.$` est dans l'unité d'organisation `Queue`.

Pour plus d'informations sur les caractères génériques dans ActiveMQ, consultezCaractères génériques

Pour fournir des règles d'autorisation pour des files d'attente spécifiques, telles que DEMO.MYQUEUE, spécifiez quelque chose comme suit :

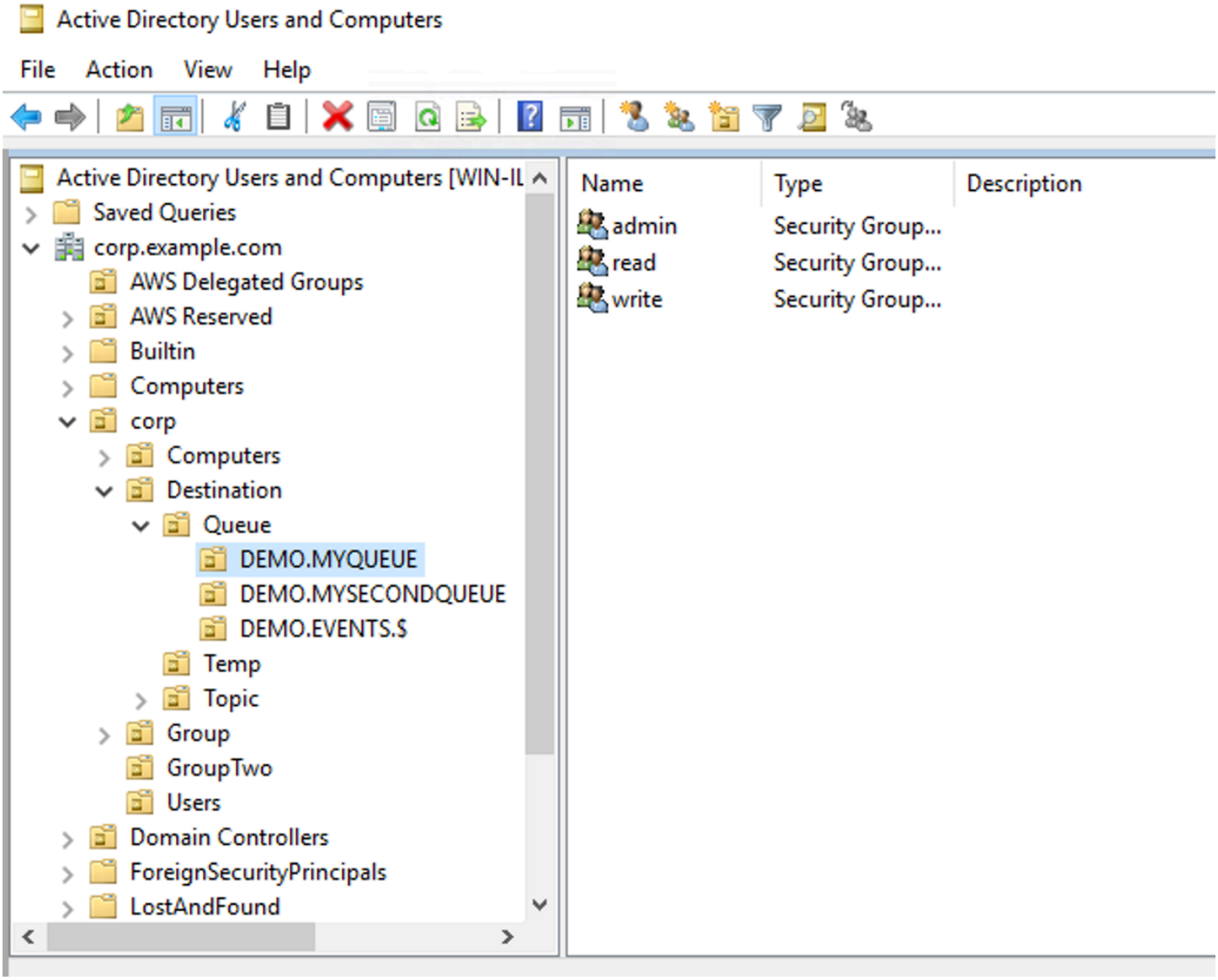
```
OU=DEMO.MYQUEUE,OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```



Groupes de sécurité

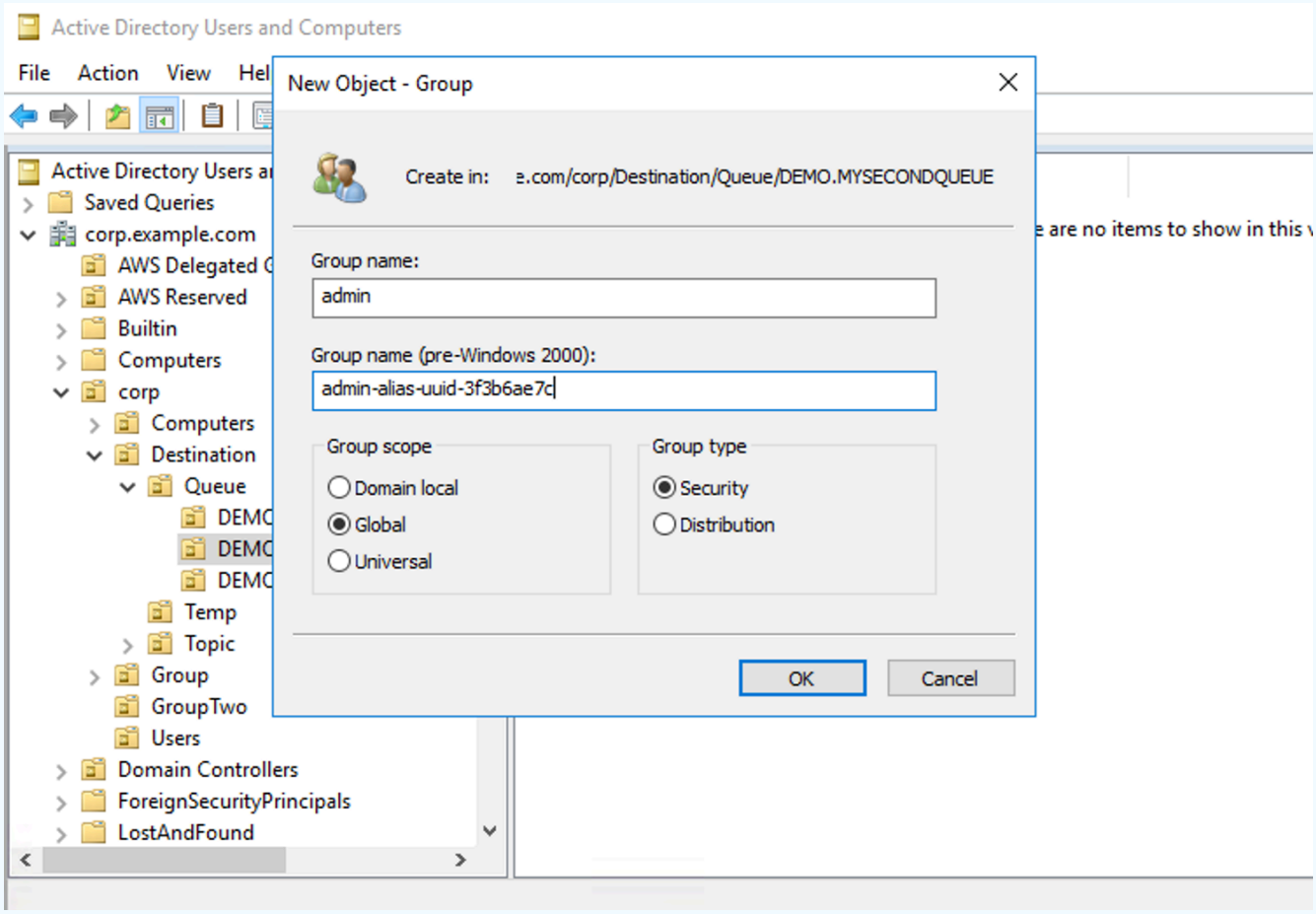
Dans chaque unité d'organisation qui représente une destination ou un caractère générique, vous devez créer trois groupes de sécurité. Comme toutes les autorisations dans ActiveMQ, il s'agit d'autorisations. read/write/admin Pour plus d'informations sur ce que chacune de ces autorisations permet à un utilisateur, consultez Sécurité dans la documentation ActiveMQ.

Vous devez nommer ces groupes de sécurité `read`, `write` et `admin`. Dans chacun de ces groupes de sécurité, vous pouvez ajouter des utilisateurs ou des groupes, qui auront ensuite l'autorisation d'effectuer les actions associées. Vous aurez besoin de ces groupes de sécurité pour chaque jeu de destinations génériques ou chaque destination individuelle.



Note

Lorsque vous créez le groupe admin, un conflit survient avec le nom du groupe. Ce conflit se produit parce que les règles antérieures à Windows 2000 héritées ne permettent pas aux groupes de partager le même nom, même si les groupes se trouvent à des emplacements différents du DIT. La valeur de la zone de texte Pre-Windows 2000 (Pré-Windows 2000) n'a aucun impact sur la configuration, mais elle doit être globalement unique. Pour éviter ce conflit, vous pouvez ajouter un suffixe `uuid` à chaque groupe `admin`.



L'ajout d'un utilisateur au groupe de sécurité `admin` pour une destination particulière permettra à l'utilisateur de créer et de supprimer cette rubrique. Les ajouter au groupe de sécurité `read` leur permettra de lire à partir de la destination, et les ajouter au groupe `write` leur permettra d'écrire dans la destination.

Outre l'ajout d'utilisateurs individuels aux autorisations de groupe de sécurité, vous pouvez également ajouter des groupes entiers. Cependant, comme ActiveMQ code à nouveau en dur les noms

d'attributs pour les groupes, vous devez vous assurer que le groupe que vous souhaitez ajouter possède la classe d'objet `groupOfNames`, comme illustré dans le code source ActiveMQ.

Pour ce faire, suivez le même processus qu'avec l'`uid` pour les utilisateurs. Consultez Configuration des mappages d'ID dans les utilisateurs et ordinateurs Active Directory pour les versions Windows Server 2016 (et ultérieures).

Étape 3 : (Facultatif) Se connecter à une AWS Lambda fonction

AWS Lambda peut se connecter à votre courtier Amazon MQ et en consommer les messages. Lorsque vous connectez un agent à Lambda, vous créez un mappage de la source d'événement qui lit les messages d'une file d'attente et appelle la fonction de manière synchrone. Le mappage de la source d'événements que vous créez lit les messages de votre agent par lots et les convertit en une charge utile Lambda sous la forme d'un objet JSON.

Pour connecter votre agent à une fonction Lambda

1. Ajoutez les autorisations de rôle IAM suivantes au rôle d'exécution de votre fonction Lambda.

   - mq : DescribeBroker
   - EC2 : CreateNetworkInterface
   - EC2 : DeleteNetworkInterface
   - EC2 : DescribeNetworkInterfaces
   - EC2 : DescribeSecurityGroups
   - EC2 : DescribeSubnets
   - EC2 : DescribeVpcs
   - journaux : CreateLogGroup
   - journaux : CreateLogStream
   - journaux : PutLogEvents
   - responsable des secrets : GetSecretValue

   > **Note**
   >
   > Sans les autorisations IAM nécessaires, votre fonction ne sera pas en mesure de lire correctement les enregistrements des ressources Amazon MQ.

2. (Facultatif) Si vous avez créé un agent sans accès public, vous devez effectuer l'une des opérations suivantes pour permettre à Lambda de se connecter à votre agent :

   - Configurez une passerelle NAT par sous-réseau public. Pour plus d'informations, consultez Accès à Internet et aux services pour les fonctions connectées à un VPC dans le Guide du développeur AWS Lambda .
   - Créez une connexion entre votre Amazon Virtual Private Cloud (Amazon VPC) et Lambda à l'aide d'un point de terminaison VPC. Votre Amazon VPC doit également se connecter à AWS Security Token Service (AWS STS) et aux points de terminaison Secrets Manager. Pour plus d'informations, consultez Configuration de points de terminaison de VPC d'interface pour Lambda dans le Guide du développeur AWS Lambda .

3. Configurez votre agent en tant que source d'événement pour une fonction Lambda à l'aide de la AWS Management Console. Vous pouvez également utiliser la `create-event-source-mapping` AWS Command Line Interface commande.

4. Écrivez du code pour votre fonction Lambda pour traiter les messages consommés par votre agent. La charge utile Lambda récupérée par votre mappage de source d'événement dépend du type de moteur de l'agent. Voici un exemple de charge utile Lambda pour une file d'attente Amazon MQ for ActiveMQ.

   > **Note**
   >
   > Dans cet exemple, `testQueue` est le nom de la file d'attente.

```
{
  "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
      }
    ]
  }
}
```

Pour plus d'informations sur la connexion d'Amazon MQ à Lambda, les options prises en charge par Lambda pour une source d'événement Amazon MQ et les erreurs de mappage de source d'événement, consultez Utilisation de Lambda avec Amazon MQ dans le Guide du développeur AWS Lambda .

Création d'un utilisateur de courtier ActiveMQ

Un utilisateur ActiveMQ est une personne ou une application qui peut accéder aux files d'attente et aux rubriques d'un agent ActiveMQ. Vous pouvez configurer les utilisateurs pour qu'ils disposent d'autorisations spécifiques. Par exemple, vous pouvez autoriser certains utilisateurs à accéder à la console web ActiveMQ.

Un groupe est une étiquette sémantique. Vous pouvez affecter un groupe à un utilisateur et configurer des autorisations pour les groupes pour envoyer vers, recevoir depuis et administrer des files d'attente et des rubriques spécifiques.

> **Note**
>
> Vous ne pouvez pas configurer des groupes indépendamment des utilisateurs. Une étiquette de groupe est créée lorsque vous ajoutez au moins un utilisateur et supprimée lorsque vous en supprimez tous les utilisateurs.

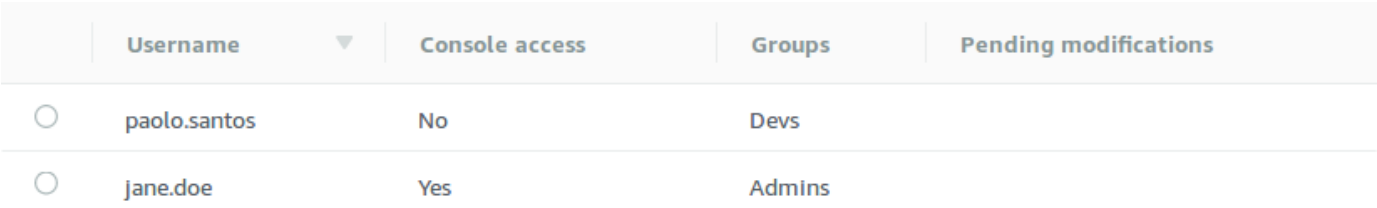
> **Note**
>
> Le `activemq-webconsole` groupe dans ActiveMQ sur Amazon MQ possède des autorisations d'administrateur sur toutes les files d'attente et sur tous les sujets. Tous les utilisateurs de ce groupe auront un accès administrateur.

Les exemples suivants montrent comment créer, modifier et supprimer des utilisateurs d'agent Amazon MQ à l'aide de AWS Management Console.

Création d'un nouvel utilisateur de courtier ActiveMQ

1. Connectez-vous à la console Amazon MQ.
2. Dans la liste des courtiers, choisissez le nom de votre courtier (par exemple MyBroker), puis choisissez Afficher les détails.

   Sur la ***MyBroker***page, dans la section Utilisateurs, tous les utilisateurs de ce courtier sont répertoriés.

3. Choisissez Create user (Créer un utilisateur).
4. Dans la boîte de dialogue Create user (Créer un utilisateur), saisissez un Username (Nom d'utilisateur) et un Password (Mot de passe).
5. (Facultatif) Saisissez les noms des groupes auxquels l'utilisateur appartient, séparés par des virgules (par exemple : `Devs, Admins`).
6. (Facultatif) Pour permettre à l'utilisateur d'accéder à la console web ActiveMQ, choisissez ActiveMQ Web Console.

7. Choisissez Create user (Créer un utilisateur).

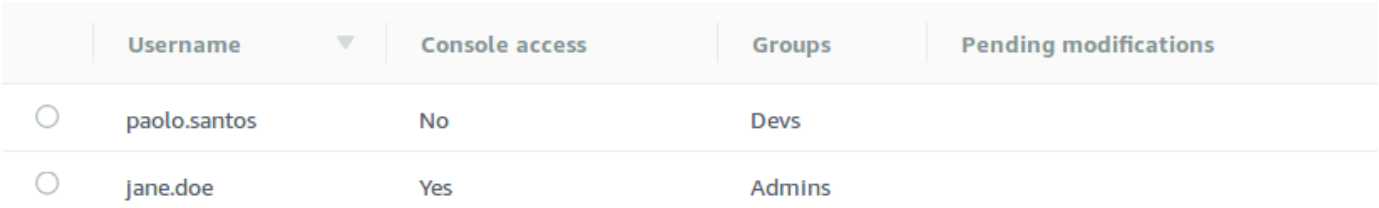
> **Important**
>
> Apporter des modifications à une configuration n'applique pas immédiatement les modifications à l'agent. Pour appliquer vos modifications, vous devez attendre la fenêtre de maintenance suivante ou redémarrer l'agent.

Modifier un utilisateur de courtier ActiveMQ

Pour modifier un utilisateur existant, procédez comme suit :

1. Connectez-vous à la console Amazon MQ.
2. Dans la liste des courtiers, choisissez le nom de votre courtier (par exemple MyBroker), puis choisissez Afficher les détails.

   Sur la ***MyBroker***page, dans la section Utilisateurs, tous les utilisateurs de ce courtier sont répertoriés.



3. Sélectionnez vos informations d'identification de connexion et choisissez Modifier.

   La boîte de dialogue Edit user (Modifier l'utilisateur) s'affiche.
4. (Facultatif) Saisissez un nouveau Password (Mot de passe).
5. (Facultatif) Ajoutez ou supprimez les noms des groupes auxquels l'utilisateur appartient, séparés par des virgules (par exemple : `Managers, Admins`).
6. (Facultatif) Pour permettre à l'utilisateur d'accéder à la console web ActiveMQ, choisissez ActiveMQ Web Console.
7. Pour enregistrer les modifications apportées à l'utilisateur, choisissez Done (Terminé).

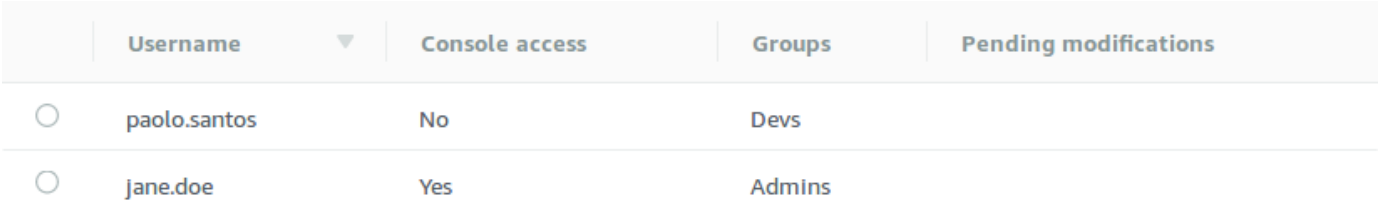
> **Important**
>
> Apporter des modifications à une configuration n'applique pas immédiatement les modifications à l'agent. Pour appliquer vos modifications, vous devez attendre la fenêtre de maintenance suivante ou redémarrer l'agent.

Supprimer un utilisateur de courtier ActiveMQ

Lorsque vous n'avez plus besoin d'un utilisateur, vous pouvez le supprimer.

1. Connectez-vous à la console Amazon MQ.
2. Dans la liste des courtiers, choisissez le nom de votre courtier (par exemple MyBroker), puis choisissez Afficher les détails.

   Sur la ***MyBroker***page, dans la section Utilisateurs, tous les utilisateurs de ce courtier sont répertoriés.



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

3. Sélectionnez vos informations de connexion (par exemple, ***MyUser***), puis choisissez Supprimer.
4. Pour confirmer la suppression de l'utilisateur, dans le champ Supprimer ***MyUser*** ? dans une boîte de dialogue, choisissez Supprimer.

> **Important**
>
> Apporter des modifications à une configuration n'applique pas immédiatement les modifications à l'agent. Pour appliquer vos modifications, vous devez attendre la fenêtre de maintenance suivante ou redémarrer l'agent.

Exemples pratiques d'utilisation de Java Message Service (JMS) avec ActiveMQ

Les exemples suivants montrent comment utiliser ActiveMQ par programmation :

- L' OpenWire exemple de code Java permet de se connecter à un courtier, de créer une file d'attente, d'envoyer et de recevoir un message. Pour obtenir une analyse et une explication détaillées, consultez Connecting a Java application to your broker.
- L'exemple de code Java MQTT se connecte à un agent, crée une rubrique, et publie et reçoit un message.
- L'exemple de code Java STOMP+WSS se connecte à un agent, crée une file d'attente, et envoie et reçoit un message.

Conditions préalables

Activer les attributs du VPC

Pour vous assurer que votre agent est accessible dans votre VPC, vous devez activer les attributs `enableDnsHostnames` et `enableDnsSupport` du VPC. Pour plus d'informations, consultez Prise en charge du DNS dans votre VPC dans le Guide de l'utilisateur Amazon VPC.

Activer les connexions entrantes

Pour utiliser Amazon MQ par programmation, vous devez utiliser des connexions entrantes.

1. Connectez-vous à la console Amazon MQ.
2. Dans la liste des courtiers, choisissez le nom de votre courtier (par exemple, MyBroker).
3. Sur la ***`MyBroker`***page, dans la section Connexions, notez les adresses et les ports de l'URL de la console Web du courtier et des protocoles au niveau du fil.
4. Dans la section Details (Détails), sous Security and network (Sécurité et réseau), choisissez le nom de votre groupe de sécurité ou

   La page Groupes de sécurité du tableau de bord EC2 est affichée.
5. Dans la liste des groupes de sécurité, choisissez votre groupe de sécurité.
6. Au bas de la page, choisissez Entrant, puis Modifier.

7. Dans la boîte de dialogue Edit inbound rules (Modifier les règles entrantes), ajoutez une règle pour chaque URL ou point de terminaison pour qu'ils soient accessibles publiquement (l'exemple suivant montre comment procéder pour une console web d'agent).

   a. Choisissez Add Rule (Ajouter une règle).

   b. Pour Type, sélectionnez Custom TCP (TCP personnalisé).

   c. Pour Port Range (Plage de ports), saisissez le port de la console web (`8162`).

   d. Pour Source, laissez l'option Custom (Personnalisée) sélectionnée, puis tapez l'adresse IP du système qui doit pouvoir accéder à la console web (par exemple, `192.0.2.1`).

   e. Choisissez Enregistrer.

      Votre agent peut désormais accepter les connexions entrantes.

Ajouter des dépendances Java

OpenWire

Ajoutez les packages `activemq-client.jar` et `activemq-pool.jar` au chemin de classe Java. L'exemple suivant illustre ces dépendances dans un fichier `pom.xml` de projet 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>
```

Pour plus d'informations sur `activemq-client.jar`, consultez Configuration initiale dans la documentation ActiveMQ Apache.

MQTT

Ajoutez le package `org.eclipse.paho.client.mqttv3.jar` au chemin de classe Java. L'exemple suivant illustre cette dépendance dans un fichier `pom.xml` de projet Maven.

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

Pour plus d'informations sur `org.eclipse.paho.client.mqttv3.jar`, consultez Eclipse Paho Java Client.

STOMP+WSS

Ajoutez les packages suivants au chemin de classe Java :

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

L'exemple suivant illustre ces dépendances dans un fichier `pom.xml` de projet 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>
```

Pour plus d'informations, consultez STOMP Support dans la documentation du framework Spring.

MQExampleAmazon.java

> **Important**
>
> Dans l'exemple de code suivant, les producteurs et les consommateurs s'exécutent dans un seul thread. Pour les systèmes de production (ou pour tester le basculement d'instance d'agent), assurez-vous que vos producteurs et vos consommateurs s'exécutent sur des hôtes ou des threads distincts.

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);
                        }
                    }
                    }
```

Gestion des versions du moteur Amazon MQ for ActiveMQ

Apache ActiveMQ organise les numéros de version en fonction de la spécification sémantique de gestion des versions comme `X.Y.Z`. Dans Amazon MQ pour les implémentations d'ActiveMQ, `X` indique la version majeure, `Y` représente la version secondaire et indique le numéro de version du correctif. `Z` Amazon MQ considère qu'une modification de version est importante si les numéros de version majeure changent. Par exemple, la mise à niveau de la version 5.17 vers la version 6.0 est considérée comme une mise à niveau de version majeure. Un changement de version est considéré

comme mineur si seul le numéro de version mineure ou de correctif change. Par exemple, mise à niveau depuis la version 5. 18 à 5 ans. La version 19 est considérée comme une mise à niveau mineure. Lorsqu'il `autoMinorVersionUpgrade` est activé, Amazon MQ met à niveau votre courtier vers la dernière version de correctif disponible.

Amazon MQ pour ActiveMQ recommande à tous les courtiers d'utiliser la dernière version mineure prise en charge. Pour obtenir des instructions sur la mise à niveau de la version de votre moteur de courtage, consultez la section Mise à niveau d'une version du moteur de courtage Amazon MQ.

Versions de moteur prises en charge sur Amazon MQ pour ActiveMQ

Le calendrier de support de la version Amazon MQ indique à quel moment le support d'une version du moteur de courtage atteindra la fin du support. Lorsqu'une version atteint la fin du support, Amazon MQ met automatiquement à niveau tous les courtiers utilisant cette version vers la version prise en charge suivante. Cette mise à niveau a lieu pendant les fenêtres de maintenance planifiées par votre courtier, dans les 45 jours suivant la end-of-support date.

Amazon MQ fournit un préavis d'au moins 90 jours avant la fin du support d'une version. Nous vous recommandons de surclasser votre courtier avant end-of-support cette date afin d'éviter toute interruption. En outre, vous ne pouvez pas créer de nouveaux courtiers sur les versions dont la fin du support est prévue dans les 30 jours suivant la date de fin du support.

Version d'Apache ActiveMQ	Fin du support sur Amazon MQ
ActiveMQ 5.19 (recommandé)	
ActiveMQ 5.18	
ActiveMQ 5.17	16 juin 2025
ActiveMQ 5.16	15 novembre 2024
ActiveMQ 5.16	16 septembre 2024

Lorsque vous créez un agent Amazon MQ for ActiveMQ, vous pouvez spécifier n'importe quelle version de moteur ActiveMQ prise en charge. Si vous ne spécifiez pas le numéro de version du moteur lors de la création d'un broker, Amazon MQ utilise automatiquement par défaut le dernier numéro de version du moteur.

Mises à niveau de la version

Vous pouvez à tout moment mettre à niveau manuellement votre courtier vers la prochaine version majeure ou mineure prise en charge. Lorsque vous activez les mises à niveau automatiques des versions mineures, Amazon MQ met à niveau votre broker vers la dernière version de correctif prise en charge pendant la période de maintenance.

Pour plus d'informations sur la mise à niveau manuelle de votre courtier, consultezthe section called “Mise à niveau de la version du moteur”.

Liste des versions de moteur prises en charge

Vous pouvez répertorier toutes les versions mineures et majeures du moteur prises en charge à l'aide de la `describe-broker-instance-options` AWS CLI commande.

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

Pour filtrer les résultats par moteur et par type d’instance, utilisez les options `--engine-type` et `--host-instance-type` comme illustré ci-dessous.

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

Par exemple, pour filtrer les résultats pour ActiveMQ et le type d'instance, *`engine-type`* remplacez `ACTIVEMQ` par `mq.m5.large` et par. *`instance-type`* `mq.m5.large`

Bonnes pratiques Amazon MQ for ActiveMQ

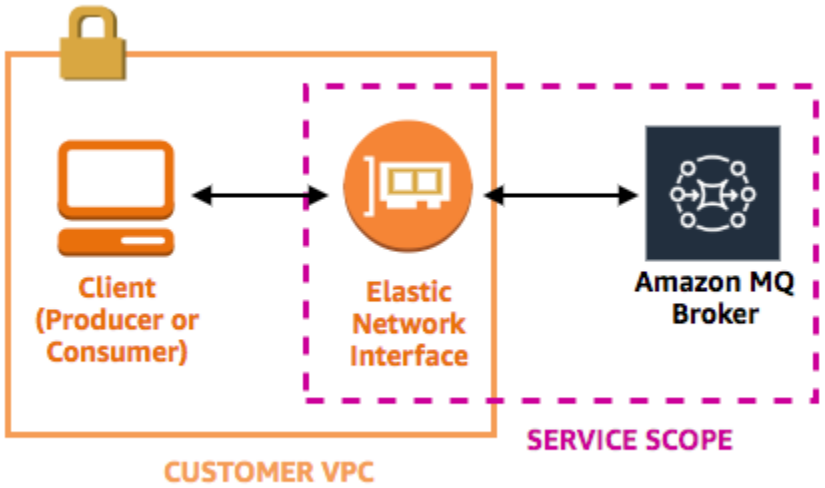
Utilisez la section comme référence pour trouver rapidement les recommandations relatives à l'optimisation des performances et à la réduction des coûts de débit lors de l'utilisation des agents ActiveMQ sur Amazon MQ.

Ne jamais modifier ou supprimer l'interface réseau Elastic Amazon MQ

Lorsque vous créez un agent Amazon MQ pour la première fois, Amazon MQ alloue une interface réseau Elastic dans le Virtual Private Cloud (VPC) sous votre compte et demande un nombre d'autorisations EC2. L'interface réseau permet à votre client (producteur ou consommateur) de

communiquer avec l'agent Amazon MQ. L'interface réseau est considérée comme étant dans la portée du service d'Amazon MQ, bien que faisant partie du VPC de votre compte.

> **Warning**
>
> Vous ne devez pas modifier ou supprimer cette interface réseau. La modification ou la suppression de l'interface réseau peut entraîner une perte définitive de la connexion entre votre VPC et votre agent.



Toujours utiliser le regroupement de connexions

Dans un scénario avec un seul producteur et un seul consommateur (comme dans le didacticielMise en route : création et connexion à un courtier ActiveMQ), vous pouvez utiliser une seule classe `ActiveMQConnectionFactory` pour chaque producteur et consommateur. Par exemple :

```
// 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();
```

Toutefois, dans des scénarios plus réalistes avec plusieurs producteurs et plusieurs consommateurs, il peut s'avérer coûteux et inefficace de créer un grand nombre de connexions pour plusieurs producteurs. Dans ces scénarios, vous devez regrouper plusieurs demandes de producteurs à l'aide de la classe `PooledConnectionFactory`. Par exemple :

> **Note**
>
> Les consommateurs de messages ne doivent jamais utiliser 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();
```

Toujours utiliser le transport de basculement pour se connecter à plusieurs points de terminaison d'agent

Si vous avez besoin que votre application se connecte à plusieurs points de terminaison d'agent, par exemple, lorsque vous utilisez un mode de déploiement actif/en veille ou lorsque vous migrez à partir d'un agent de messages sur site vers Amazon MQ, utilisez le transport de basculement pour autoriser vos consommateurs à se connecter de façon aléatoire à l'un des points de terminaison. Par exemple :

```
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**
>
> Les courtiers en zones de multidisponibilité peuvent subir des défaillances lors des fenêtres de maintenance et des redémarrages des courtiers. Utilisez le Failover Transport pour garantir la disponibilité de votre courtier.

Éviter d'utiliser des sélecteurs de messages

Il est possible d'utiliser des sélecteurs JMS pour attacher des filtres à des inscriptions à des rubriques (pour acheminer des messages vers des consommateurs en fonction de leur contenu). Toutefois, l'utilisation des sélecteurs JMS remplit la mémoire tampon du filtre de l'agent Amazon MQ, ce qui empêche le filtrage des messages.

En général, évitez de laisser les consommateurs acheminer des messages car, pour un découplage optimal des consommateurs et des producteurs, le consommateur et le producteur doivent être éphémères.

Préférer des destinations virtuelles à des abonnements durables

Un abonnement durable permet de s'assurer que le consommateur reçoit tous les messages publiés dans une rubrique, par exemple, après la restauration d'une connexion perdue. Cependant, l'utilisation d'abonnements durables interdit également l'utilisation des consommateurs concurrents et peut donner lieu à des problèmes à grande échelle. Envisagez d'utiliser plutôt des destinations virtuelles.

Si vous utilisez le peering Amazon VPC, évitez les clients IPs dans la plage d'adresses CIDR **`10.0.0.0/16`**

Si vous configurez le peering Amazon VPC entre une infrastructure sur site et votre courtier Amazon MQ, vous ne devez pas configurer de connexions client dans une plage d'adresses CIDR. IPs `10.0.0.0/16`

Désactiver Concurrent Store and Dispatch (Répartition et stockage simultanés) pour les files d'attente à consommateurs lents

Par défaut, Amazon MQ est optimisé pour les files d'attente avec consommateurs rapides :

- Des consommateurs sont considérées comme étant rapides s'ils peuvent suivre le débit des messages générés par des producteurs.
- Des consommateurs sont considérés comme étant lents si une liste d'attente de messages non reconnus est créé dans une file d'attente, ce qui peut potentiellement entraîner une baisse du débit du producteur.

Pour demander à Amazon MQ d'être optimisé pour les files d'attente avec des consommateurs lents, définissez l'attribut `concurrentStoreAndDispatchQueues` sur `false`. Pour accéder à un exemple de configuration, consultez `concurrentStoreAndDispatchQueues`.

Choisir le type d'instance d'agent adéquat pour un débit optimal

Le débit de message d'un type d'instance d'agent dépend du cas d'utilisation de votre application et des facteurs suivants :

- Utilisation d'ActiveMQ en mode persistant
- Message size (Taille de message)
- Nombre de producteurs et de consommateurs
- Numéro de la destination

Compréhension de la relation entre la taille du message, la latence et le débit

Selon votre cas d'utilisation, un type d'instance d'agent plus grand pourrait ne pas améliorer le débit du système. Lorsqu'ActiveMQ écrit des messages pour un stockage durable, la taille de vos messages détermine la limite de votre système :

- Si la taille de vos messages est inférieure à 100 Ko, la latence de stockage permanent représente la limite.
- Si la taille de vos messages est supérieure à 100 Ko, le débit de stockage permanent représente la limite.

Lorsque vous utilisez ActiveMQ en mode persistant, l'écriture en stockage s'effectue normalement lorsqu'il y a peu de consommateurs ou lorsque les consommateurs sont lents. En mode non persistant, l'écriture en stockage s'effectue aussi avec des consommateurs lents si la mémoire du segment de l'instance d'agent est pleine.

Pour déterminer le meilleur type d'instance d'agent pour votre application, nous vous recommandons de tester différents types d'instance d'agent. Pour plus d'informations, consultez Broker instance types et Mesurer le débit pour Amazon MQ à l'aide de l'évaluation JMS.

Cas d'utilisation pour les grands types d'instance d'agent

Il existe trois cas d'utilisation courants où des types d'instance d'agent plus grands améliore le débit :

- Non-persistent mode (Mode non persistant) : Lorsque votre application est moins sensible à la perte de messages pendant le basculement d'une instance d'agent (par exemple, lors de la diffusion du score d'un sport), vous pouvez la plupart du temps utiliser ActiveMQ en mode non persistant. Dans ce mode, ActiveMQ écrit des messages pour un stockage permanent uniquement si la mémoire du segment de l'instance d'agent est pleine. Les systèmes qui utilise le mode non persistant peuvent bénéficier d'un volume de mémoire plus important, d'un processeur plus rapide et d'un réseau plus rapide disponible sur des types d'instance d'agent plus grands.
- Fast consumers (Consommateurs rapides) : Lorsque les consommateurs actifs sont disponibles et que l'indicateur `concurrentStoreAndDispatchQueues` est activé, ActiveMQ autorise les messages à transiter directement des producteurs aux consommateurs sans envoyer de messages au stockage (même en mode persistant). Si votre application peut consommer des messages rapidement (ou si vous pouvez attribuer ce rôle à vos consommateurs), elle peut bénéficier d'un type d'instance d'agent plus grand. Pour permettre à votre application de consommer des messages plus rapidement, ajoutez des threads de consommateur aux instances de votre application, ou augmentez la taille de votre application verticalement ou horizontalement.
- Batched transactions (Transactions par lot) : Lorsque vous utilisez le mode persistant et envoyez plusieurs messages par transaction, vous pouvez obtenir un débit global plus élevé de messages en utilisant des types d'instance d'agent plus grands. Pour plus d'informations, consultez Should I Use Transactions? dans la documentation ActiveMQ Apache.

Choisir le type de stockage d'agent adéquat pour un débit optimal

Pour tirer parti d'une grande durabilité et d'une réplication sur plusieurs zones de disponibilité, utilisez Amazon EFS. Pour profiter d'une faible latence et d'un débit élevé, utilisez Amazon EBS. Pour de plus amples informations, veuillez consulter Storage.

Correctement configurer votre réseau d'agents

Lorsque vous créez un réseau d'agents, configurez-le correctement pour votre application :

- Enable persistent mode (Activer le mode persistant) : Puisque (par rapport à ses homologues) chaque instance de l'agent fonctionne comme un producteur ou un consommateur, les réseaux d'agents ne fournissent pas de réplication de messages distribuée. Le premier agent qui agit en tant que consommateur reçoit un message et le conserve pour le stockage. Cet agent envoie un accusé de réception au producteur et transmet le message à l'agent suivant. Lorsque le deuxième agent confirme la persistance du message, le premier agent supprime le message.

  Si le mode persistant est désactivé, le premier agent envoie un accusé de réception au producteur sans conserver le message pour le stockage. Pour plus d'informations, consultez les sections Replicated Message Store (Stockage de messages répliqués) et What is the difference between persistent and non-persistent delivery? (Quelle est la différence entre une livraison persistante et non persistante ?) dans la documentation Apache ActiveMQ.
- Don't disable advisory messages for broker instances (Ne pas désactiver les messages consultatifs pour les instances d'agent) : Pour plus d'informations, consultez la section Advisory Message (Message consultatif) dans la documentation Apache ActiveMQ.
- Don't use multicast broker discovery (Ne pas utiliser de détection d'agent à multidiffusion) : Amazon MQ ne prend pas en charge la détection d'agent à l'aide de la multidiffusion. Pour plus d'informations, consultez What is the difference between discovery, multicast, and zeroconf? (Quelle est la différence entre la détection, la multidiffusion et la zeroconf ?) dans la documentation Apache ActiveMQ.

Éviter les redémarrages lents en récupérant des transactions XA préparées

ActiveMQ prend en charge les transactions distribuées (XA). Savoir comment ActiveMQ traite les transactions XA peut vous aider à éviter des durées de récupération excessives pour les redémarrages et les basculements d'agent Amazon MQ

Les transactions XA préparées non résolues sont réutilisées à chaque redémarrage. Si celles-ci restent non résolues, leur nombre augmente au fil du temps, ce qui rallonge considérablement le temps nécessaire pour démarrer l'agent. Ceci affecte les temps de redémarrage et de basculement. Vous devez résoudre ces transactions avec une opération `commit()` ou `rollback()` pour que les performances ne se dégradent pas au fil du temps.

Pour surveiller vos transactions XA préparées non résolues, vous pouvez utiliser la `JournalFilesForFastRecovery` métrique dans Amazon CloudWatch Logs. Si ce nombre augmente, ou est constamment supérieur à 1, vous devez récupérer vos transactions non résolues avec un code similaire à l'exemple suivant. Pour de plus amples informations, veuillez consulter Quotas dans Amazon MQ.

L'exemple de code suivant vérifie les transactions XA préparées et les ferme avec une opération `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) {
        }
    }
}
```

Dans un scénario réel, vous pouvez vérifier vos transactions XA préparées dans votre gestionnaire de transaction XA. Vous pouvez ensuite choisir de gérer chaque transaction préparée avec une opération `rollback()` ou `commit()`.

Utilisation d'Amazon MQ pour RabbitMQ

Amazon MQ facilite la création d'un agent de messages avec les ressources de calcul et de stockage adaptées à vos besoins. Vous pouvez créer, gérer et supprimer des courtiers à l'aide de l' AWS Management Console API REST Amazon MQ ou du. AWS Command Line Interface

Cette section décrit les éléments de base d'un agent de messages pour les types de moteurs ActiveMQ et RabbitMQ, répertorie les types d'instance d'agent Amazon MQ disponibles ainsi que leur état, et présente l'architecture d'un agent et les options de configuration d'un agent.

Pour en savoir plus sur Amazon MQ REST APIs, consultez le manuel Amazon MQ REST API Reference.

Qu'est-ce qu'un courtier Amazon MQ pour RabbitMQ ?

Un agent est un environnement d'agent de messages qui s'exécute sur Amazon MQ. Il constitue la composante de base d'Amazon MQ. La description combinée de la classe d'instance de courtier (`m7g`) et de la taille (`large`,`medium`) est appelée type d'instance de courtier (par exemple,`mq.m7g.large`).

- Un broker à instance unique se compose d'un courtier dans une zone de disponibilité derrière un Network Load Balancer (NLB). L'agent communique avec votre application et avec un volume de stockage Amazon EBS.
- Un déploiement en cluster est un regroupement logique de trois nœuds d'agent RabbitMQ derrière un dispositif d'équilibrage de charge de réseau, chacun partageant des utilisateurs, des files d'attente et un état distribué sur plusieurs zones de disponibilité (AZ).

Pour plus d'informations, voir Déploiement d'un courtier RabbitMQ.

Ports d'écouteur

Les courtiers RabbitMQ gérés par Amazon MQ prennent en charge les ports d'écoute suivants pour la connectivité au niveau de l'application via. `amqps` Vous pouvez également utiliser ces ports pour les connexions client à l'aide de la console Web RabbitMQ et de l'API de gestion. Toutes les connexions utilisent le cryptage TLS pour des raisons de sécurité.

- Port d'écoute `5671` : utilisé pour les connexions AMQP sécurisées établies via l'URL AMQP sécurisée. Ce port prend en charge les protocoles AMQP 0-9-1 et AMQP 1.0

dans RabbitMQ 4. Par exemple, étant donné un agent avec un ID d'agent `b-c8352341-ec91-4a78-ad9c-a43f23d325bb`, déployé dans la région `us-west-2`, ce qui suit est l'URL `amqps` complète de l'agent : `b-c8352341-ec91-4a78-ad9c-a43f23d325bb.mq.us-west-2.amazonaws.com:5671`.

- Ports d'écoute `443` et `15671` - Vous pouvez utiliser les deux ports d'écoute de manière interchangeable pour accéder à un courtier via la console Web RabbitMQ ou l'API de gestion. Le port 443 fournit un accès HTTPS standard, tandis que le port 15671 est le port de gestion RabbitMQ traditionnel avec cryptage TLS.

Attributes

Un agent RabbitMQ a plusieurs attributs :

- Un nom Par exemple, `MyBroker`.
- Un ID Par exemple, `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
- Un Amazon Resource Name (ARN) Par exemple, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
- Une URL de la console web RabbitMQ. Par exemple, `https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com`.

  Pour plus d'informations, consultez Console web RabbitMQ dans la documentation RabbitMQ.
- Un point de terminaison AMQP sécurisé. Par exemple, `amqps://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com`.

Pour obtenir la liste complète des attributs des agents, consultez ce qui suit dans la référence d'API REST Amazon MQ :

- ID d'opération REST : Agent
- ID d'opération REST : Agents
- ID d'opération REST : Redémarrage d'agent

Gestion des versions du moteur Amazon MQ for RabbitMQ

RabbitMQ organise les numéros de version en fonction de la spécification sémantique de gestion des versions comme `X.Y.Z`. Dans Amazon MQ pour les implémentations de RabbitMQ, `X` indique

la version majeure, Y représente la version mineure et indique le numéro de version du correctif. Z Amazon MQ considère qu'une modification de version est importante si les numéros de version majeure changent. Par exemple, la mise à niveau de la version 3.13 vers la version 4.0 est considérée comme une mise à niveau de version majeure. Un changement de version est considéré comme mineur si seul le numéro de version mineure ou de correctif change. Par exemple, mise à niveau depuis la version 3. 11,28 à 3. La version 12.13 est considérée comme une mise à niveau mineure.

Amazon MQ pour RabbitMQ recommande à tous les courtiers d'utiliser la dernière version prise en charge de RabbitMQ 4.2. Pour obtenir des instructions sur la mise à niveau de la version de votre moteur de courtage, consultez la section Mise à niveau d'une version du moteur de courtage Amazon MQ.

Lorsque vous créez un nouveau courtier Amazon MQ pour RabbitMQ, il vous suffit de spécifier les numéros de version principale et secondaire. Par exemple, RabbitMQ 4.2. Si vous ne spécifiez pas la version du moteur lors de la création d'un broker, Amazon MQ utilise automatiquement par défaut la dernière version du moteur.

> **Important**
>
> Amazon MQ ne prend pas en charge les flux. La création d'un flux entraînera une perte de données.
> Amazon MQ ne prend pas en charge l'utilisation de la journalisation structurée en JSON.

Amazon MQ prend en charge deux versions majeures de RabbitMQ :

- Lapin MQ 4

  Amazon MQ prend en charge RabbitMQ 4.2 dans la série de versions RabbitMQ 4 uniquement sur le type d'instance mq.m7g, quelle que soit la taille d'instance prise en charge.

- RabbitMQ 3

  Amazon MQ prend en charge RabbitMQ 3.13 dans la série de versions RabbitMQ 3 sur les types d'instances mq.t3, mq.m5 et mq.m7g, quelle que soit la taille d'instance prise en charge.

Liste des versions de moteur prises en charge

Vous pouvez répertorier toutes les versions mineures et majeures du moteur prises en charge à l'aide de la `describe-broker-instance-options` AWS CLI commande.

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

Pour filtrer les résultats par moteur et par type d'instance, utilisez les options `--engine-type` et `--host-instance-type` comme illustré ci-dessous.

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

Par exemple, pour filtrer les résultats pour RabbitMQ et le type d'`mq.m7g.large`instance, remplacez par et *`engine-type`* par`RABBITMQ`. *`instance-type`* `mq.m7g.large`

Lapin MQ 4

Amazon MQ prend en charge RabbitMQ 4.2 dans la série de versions RabbitMQ 4 uniquement sur le type d'instance mq.m7g, quelle que soit la taille d'instance prise en charge.

> **Important**
>
> Vous ne pouvez créer de nouveaux courtiers que sur RabbitMQ 4.2. Les mises à niveau en place depuis RabbitMQ 3.13 ne sont actuellement pas prises en charge.

> **Important**
>
> Le type de file d'attente par défaut sur Amazon MQ pour les courtiers RabbitMQ 4.2 sera « quorum ». Si aucun argument de type de file d'attente n'est spécifié lors de la création de la file, une file d'attente quorum sera créée.
> Nous recommandons vivement d'utiliser des files d'attente de quorum sur RabbitMQ 4 pour des raisons de durabilité, car la durabilité des files d'attente classiques n'est pas garantie dans tous les cas.

Les modifications suivantes ont été introduites dans RabbitMQ 4 sur Amazon MQ

- AMQP 1.0 en tant que protocole de base : pour plus d'informations, voir Protocoles.
- Pelles locales : les pelles supportent désormais un nouveau protocole appelé « local » en plus des protocoles AMQP 0-9-1 et AMQP 1.0. Les pelles locales sont basées en interne sur l'AMQP 1.0, mais au lieu d'utiliser des connexions TCP distinctes, elles utilisent des connexions intra-cluster entre les nœuds du cluster et des connexions internes APIs pour publier et consommer des messages. Cela ne peut être utilisé que pour la consommation et la publication au sein du même cluster et peut offrir un débit plus élevé tout en utilisant moins de ressources que les AMQP 0-9-1 et AMQP 1.0.
- Les files d'attente de quorum prennent en charge les priorités des messages : les priorités des messages de la file d'attente de quorum sont toujours actives et ne nécessitent aucune politique pour fonctionner. Dès qu'une file d'attente de quorum reçoit un message avec une priorité définie, elle active la priorisation. Les files d'attente internes pour le quorum ne prennent en charge que deux priorités : haute et normale. Les messages sans définition de priorités seront mappés à la normale, tout comme les priorités 0 à 4. Les messages dont la priorité est supérieure à 4 seront mappés sur une priorité élevée. Les messages à priorité élevée seront privilégiés par rapport aux messages à priorité normale selon un ratio de 2:1, c'est-à-dire que pour 2 messages de priorité élevée, la file d'attente délivrera un message de priorité normale (si disponible). Par conséquent, les files d'attente pour le quorum mettent en œuvre une sorte de traitement prioritaire non strict et « équitable ». Cela garantit que des progrès sont toujours réalisés sur les messages prioritaires normaux, mais que les priorités élevées sont privilégiées dans un ratio de 2:1.
- Khepri : Khepri est utilisé comme magasin de métadonnées par défaut pour les courtiers RabbitMQ 4
- TLS mutuel (mTLS) : Amazon MQ prend en charge le protocole TLS mutuel (MTL) pour les courtiers RabbitMQ, permettant aux clients de s'authentifier à l'aide de certificats. Pour plus d'informations, consultez la section Configuration de mTLS.
- Plug-in d'authentification par certificat SSL : le plug-in d'authentification SSL utilise des certificats clients issus de connexions mTLS pour authentifier les utilisateurs, permettant ainsi l'authentification à l'aide de certificats clients X.509 au lieu des informations d'identification par nom d'utilisateur et mot de passe. Pour plus d'informations, consultez la section Authentification par certificat SSL.
- Plug-in d'authentification HTTP : Le plugin principal d'authentification HTTP permet de déléguer l'authentification et l'autorisation à un service HTTP externe. Pour plus d'informations, consultez Authentification et autorisation HTTP.

- Support JMS : le courtier prend désormais en charge les charges de travail JMS avec le plugin d'échange de sujets JMS activé, permettant aux applications JMS de se connecter à l'aide du client JMS RabbitMQ.

Les fonctionnalités suivantes ont été supprimées de RabbitMQ 4 sur Amazon MQ

- Mise en miroir des files d'attente classiques : les files d'attente classiques continuent d'être prises en charge sans aucune modification majeure pour les bibliothèques clientes et les applications, mais il s'agit désormais d'un type de file d'attente non répliqué. Les clients pourront se connecter à n'importe quel nœud pour publier et consommer depuis n'importe quelle file d'attente classique non répliquée. Les files d'attente de quorum sont recommandées pour la réplication et la sécurité des données.
- Suppression de la QoS globale : il est recommandé aux clients de définir la QoS par consommateur (non globale) au lieu de la QoS globale, où un seul prefetch partagé est utilisé pour un canal entier.
- Support pour les files d'attente transitoires et non exclusives : les files d'attente transitoires sont des files d'attente dont la durée de vie est liée au temps de disponibilité du nœud sur lequel elles sont déclarées. Dans un broker à instance unique, ils sont supprimés lorsque le nœud est redémarré. Dans un déploiement en cluster, ils sont supprimés lorsque le nœud sur lequel ils sont hébergés est redémarré. Nous vous recommandons d'utiliser le TTL de file d'attente pour supprimer automatiquement les files d'attente inutilisées et inactives après un certain temps d'inactivité. Les files d'attente exclusives continuent d'être prises en charge et sont supprimées une fois que toutes les connexions à la file d'attente ont été supprimées.

Les modifications majeures suivantes peuvent avoir un impact sur vos applications lors de la mise à niveau vers RabbitMQ 4.2 sur Amazon MQ

- Type de file d'attente par défaut : Le type de file d'attente par défaut sur un broker RabbitMQ 4 est défini sur quorum. Si aucun argument de type de file d'attente n'est spécifié lors de la création de la file, une file d'attente quorum sera créée.
- La limite de rediffusion par défaut pour les files d'attente du quorum est fixée à 20 : les messages redistribués 20 fois ou plus seront mis en échec ou supprimés (supprimés). Si 20 livraisons par message constituent un scénario courant pour une file d'attente, une cible de mise en attente ou une limite supérieure doit être configurée pour ces files d'attente afin d'éviter toute perte de données. La méthode recommandée pour ce faire est d'établir une politique.

- amqplib : Les versions du client Node JS amqplib antérieures à 0.10.7 ou toute bibliothèque cliente AMQP utilisant frame_max < 8192 ne pourront pas se connecter à RabbitMQ
- Limites de ressources par défaut : Amazon MQ pour RabbitMQ a introduit des limites d'utilisation des ressources par défaut pour les connexions, les canaux, les consommateurs par canal, les files d'attente, les hôtes virtuels, les pelles, les échanges et la taille maximale des messages. Ils servent de garde-fous pour protéger la disponibilité des courtiers et peuvent être personnalisés à l'aide de configurations adaptées à vos besoins spécifiques.

Les fonctionnalités suivantes ne sont pas prises en charge sur RabbitMQ 4 sur Amazon MQ

- Échanges aléatoires locaux : les échanges aléatoires locaux ne sont pas pris en charge sur Amazon MQ car les nœuds Amazon MQ se trouvent derrière un équilibreur de charge réseau.
- Intercepteur de messages : les intercepteurs de messages RabbitMQ ne sont pas pris en charge sur Amazon MQ.
- Statistiques par file d'attente : Amazon MQ ne vendra pas les métriques de file d'attente de RabbitMQ aux courtiers de RabbitMQ 4 par l'intermédiaire de RabbitMQ 4. AWS CloudWatch Amazon MQ continuera de fournir des statistiques au niveau des courtiers via. AWS CloudWatch Vous pouvez interroger les métriques de file d'attente à l'aide de l'API de gestion RabbitMQ. Nous vous recommandons d'interroger les métriques relatives à des files d'attente spécifiques à une fréquence d'une minute ou à des intervalles plus longs.

Support de version RabbitMQ

Le calendrier de support de la version Amazon MQ ci-dessous indique à quel moment le support d'une version du moteur de courtage atteindra la fin du support. Lorsqu'une version atteint la fin du support, Amazon MQ met automatiquement à niveau tous les courtiers utilisant cette version vers la version prise en charge suivante. Cette mise à niveau a lieu pendant les périodes de maintenance planifiées par votre courtier, dans les 45 jours suivant la end-of-support date.

Amazon MQ fournit un préavis d'au moins 90 jours avant la fin du support d'une version. Nous vous recommandons de surclasser votre courtier avant end-of-support cette date afin d'éviter toute interruption. En outre, vous ne pouvez pas créer de nouveaux courtiers sur les versions dont la fin du support est prévue dans les 30 jours suivant la date de fin du support.

Version RabbitMQ	Fin du support sur Amazon MQ
4.2 (recommandé)	
3.13	
3,12	17 mars 2025

Améliorations de version

Vous pouvez à tout moment mettre à niveau manuellement votre courtier vers la prochaine version majeure ou mineure prise en charge. Pour plus d'informations sur la mise à niveau manuelle de votre courtier, consultez la section Mise à niveau d'une version du moteur de courtage Amazon MQ.

Amazon MQ gère les mises à niveau vers la dernière version de correctif prise en charge pour tous les courtiers RabbitMQ utilisant la version 3.13 ou supérieure. Les mises à niveau des versions manuelles et automatiques se produisent pendant la fenêtre de maintenance planifiée ou après le redémarrage de votre agent.

> **Important**
>
> RabbitMQ autorise uniquement les mises à jour de version incrémentielles (par ex. : 3.9.x vers 3.10.x). Vous ne pouvez pas ignorer les versions mineures lors de la mise à jour (ex : 3.8.x vers 3.11.x).

Les agents à instance unique seront hors ligne lors de leur redémarrage. Pour les courtiers de clusters, les files d'attente en miroir doivent être synchronisées lors du redémarrage. Avec des files d'attente plus longues, le processus de synchronisation des files d'attente peut prendre plus de temps. Pendant le processus de synchronisation des files d'attente, la file d'attente n'est pas disponible pour les consommateurs et les producteurs. Lorsque le processus de synchronisation des files d'attente est terminé, le broker redevient disponible. Pour minimiser l'impact, nous vous recommandons de procéder à une mise à niveau en période de faible trafic. Pour plus d'informations sur les meilleures pratiques relatives aux mises à niveau de version, consultezBonnes pratiques Amazon MQ for RabbitMQ.

Options de déploiement pour Amazon MQ pour les courtiers RabbitMQ

Les agents RabbitMQ peuvent être créés en tant qu'agent à instance unique ou dans un déploiement en cluster. Pour les deux modes de déploiement, Amazon MQ offre une durabilité élevée en stockant ses données de manière redondante.

Vous pouvez accéder à vos agents RabbitMQ via tout langage de programmation pris en charge par RabbitMQ et en activant explicitement TLS pour les protocoles suivants :

- AMQP (0-9-1)

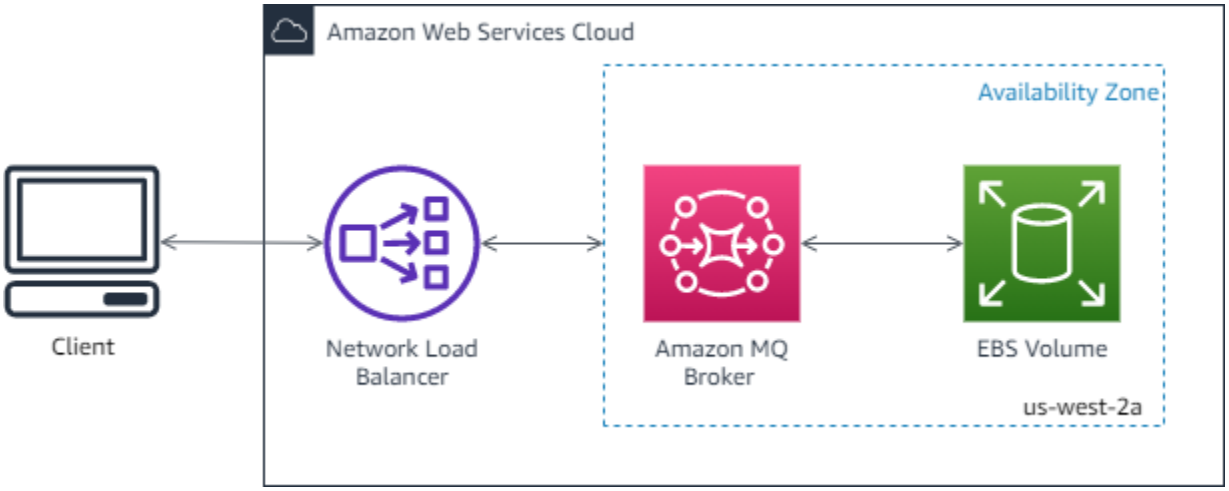
Rubriques

Option 1 : courtier à instance unique Amazon MQ pour RabbitMQ

Un agent à instance unique est composé d'un agent dans une zone de disponibilité derrière un dispositif d'équilibrage de charge de réseau. L'agent communique avec votre application et avec un volume de stockage Amazon EBS. Amazon EBS fournit un stockage de niveau bloc optimisé pour une faible latence et un débit élevé.

L'utilisation d'un Network Load Balancer garantit que le point de terminaison de votre courtier Amazon MQ pour RabbitMQ reste inchangé si l'instance de courtier est remplacée pendant une période de maintenance ou en raison de défaillances matérielles Amazon sous-jacentes. EC2 Un dispositif d'équilibrage de charge de réseau permet à vos applications et utilisateurs de continuer à utiliser le même point de terminaison pour se connecter à l'agent.

Le diagramme suivant illustre un agent à instance unique Amazon MQ for RabbitMQ.



Option 2 : déploiement du cluster Amazon MQ pour RabbitMQ

Un déploiement en cluster est un regroupement logique de trois nœuds d'agent RabbitMQ derrière un dispositif d'équilibrage de charge de réseau, chacun partageant des utilisateurs, des files d'attente et un état distribué sur plusieurs zones de disponibilité (AZ).

Dans un déploiement en cluster, Amazon MQ gère automatiquement les politiques d'agent pour activer la mise en miroir classique sur tous les nœuds, garantissant ainsi une haute disponibilité (HA). Chaque file d'attente en miroir se compose d'un nœud principal et d'un ou plusieurs miroirs. Chaque file d'attente a son propre nœud principal. Toutes les opérations d'une file d'attente donnée sont d'abord appliquées sur le nœud principal de la file d'attente, puis propagées aux miroirs. Amazon MQ crée une politique système par défaut qui définit `ha-mode` sur `all` et `ha-sync-mode` sur `automatic`. Cela garantit que les données sont répliquées sur tous les nœuds du cluster dans différentes zones de disponibilité pour une meilleure durabilité.

> **Note**
>
> Lors d'un déploiement en cluster, en cas de panne de la zone de disponibilité, Amazon MQ tentera automatiquement de déplacer les nœuds RabbitMQ concernés vers une autre zone de disponibilité afin de maintenir la taille du cluster. Une fois la panne résolue, le cluster sera automatiquement rééquilibré sur l'ensemble du AZs.

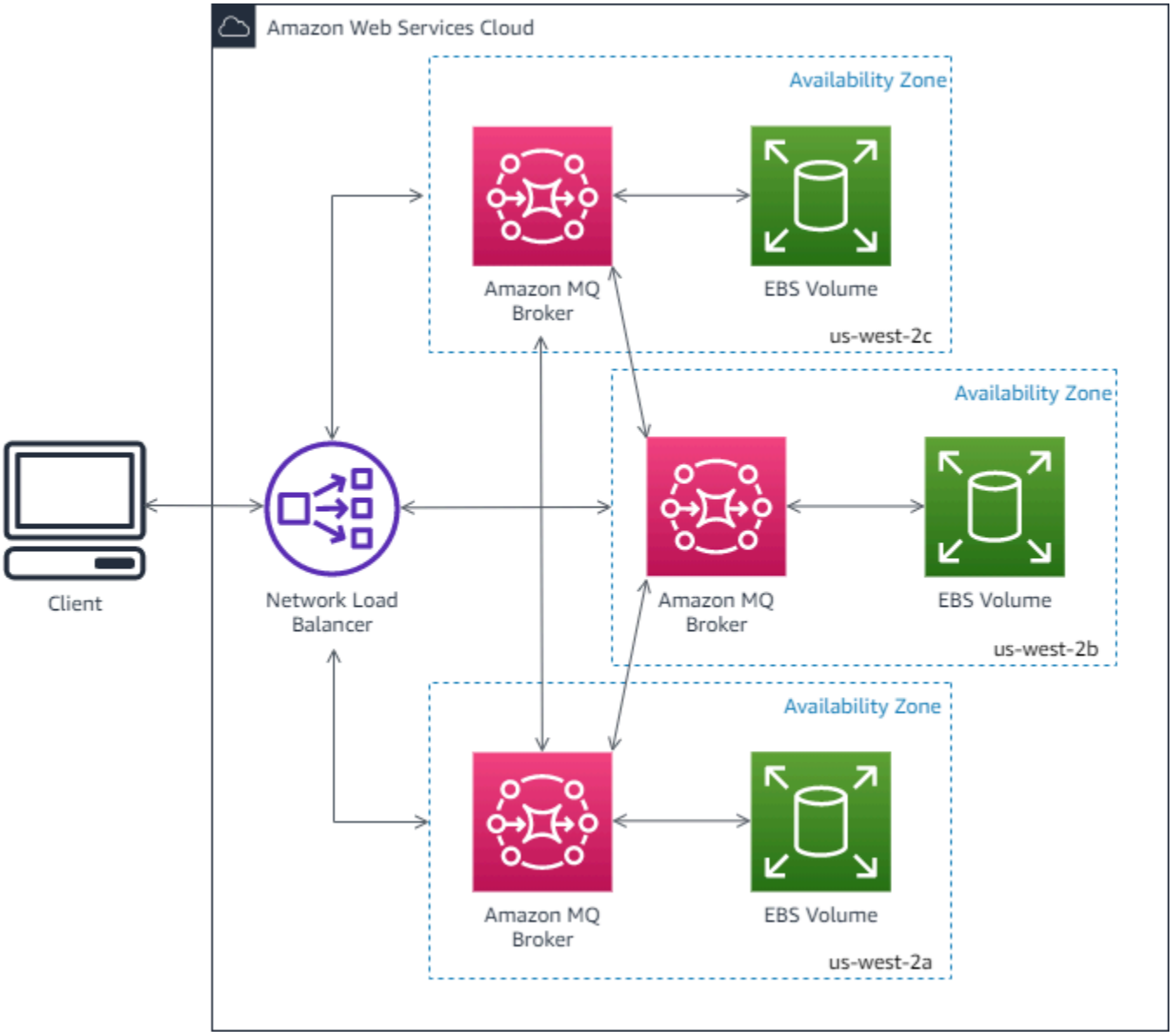
> **Note**
>
> Lors d'une fenêtre de maintenance, toute la maintenance d'un cluster est effectuée un nœud à la fois, gardant au moins deux nœuds en cours d'exécution à tout moment. Chaque

fois qu'un nœud est arrêté, les connexions client à ce nœud sont coupées et doivent être rétablies. Vous devez vous assurer que votre code client est conçu pour se reconnecter automatiquement à votre cluster. Pour plus d'informations sur la connectivité à distance, consultez the section called “Étape 1 : restauration automatique en cas de défaillance du réseau”.

Parce qu'Amazon MQ définit `ha-sync-mode: automatic` au cours d'une fenêtre de maintenance, les files d'attente se synchronisent lorsque chaque nœud rejoint le cluster. La synchronisation de la file d'attente bloque toutes les autres opérations de file Vous pouvez atténuer l'impact de la synchronisation des files d'attente pendant les fenêtres de maintenance en gardant les files d'attente courtes.

La politique par défaut ne doit pas être supprimée. Si vous supprimez cette politique, Amazon MQ la recréera automatiquement. Amazon MQ s'assure également que les propriétés de haute disponibilité sont appliquées à toutes les autres politiques que vous créez sur un agent en cluster. Si vous ajoutez une politique sans les propriétés de haute disponibilité, Amazon MQ les ajoutera pour vous. Si vous ajoutez une politique avec différentes propriétés de haute disponibilité, Amazon MQ les remplacera. Pour plus d'informations sur la mise en miroir classique, consultez Files d'attente classiques mises en miroir.

Le diagramme suivant illustre un déploiement d'agent en cluster RabbitMQ avec trois nœuds dans trois zones de disponibilité (AZ), chacun ayant son propre volume Amazon EBS et un état partagé. Amazon EBS fournit un stockage de niveau bloc optimisé pour une faible latence et un débit élevé.



Types d'instances de courtier Amazon MQ pour RabbitMQ

La description combinée de la classe d'instance de courtier (m7g) et de la taille (grande, moyenne) est appelée le type d'instance de courtier (par exemple, mq.m7g.large).

Nous recommandons d'utiliser les types d'instance mq.m7g pour les déploiements en cluster et en instance unique.

Amazon MQ fournit un préavis d'au moins 90 jours avant la fin du support d'un type d'instance. Nous vous recommandons de mettre à niveau votre courtier vers un nouveau type d'instance avant end-of-support cette date afin d'éviter toute interruption.

> **Important**
>
> Vous ne pouvez pas rétrograder un broker d'un type d'`mq.m5`instance `mq.m7g` ou vers un type d'`mq.t3.micro`instance.
> Le type d'`mq.t3.micro`instance ne prend pas en charge le déploiement en cluster.

Types d'instances pour le déploiement de clusters m7g

Nous recommandons d'utiliser des types d'`mq.m7g.x`instances pour le déploiement en cluster. Le tableau suivant indique les types d'`mq.m7g.x`instances disponibles pour le déploiement de clusters.

Type d'instance	vCPU	Mémoire (Gio)	Ligne de base du réseau/ bande passante en rafale (Gbit/s)	Usage recommandé	Stockage	Taille du volume de disque par nœud (Go)
mq.m7g.medium	1	4	0,52 / 12,5	Évaluation	EBS	5
mq.m7g.large	2	8	0,937 / 12,5	Production	EBS	15
mq.m7g.xlarge	4	16	1,876/ 12,5	Production	EBS	25
mq.m7g.2xlarge	8	32	3,75 / 15,0	Production	EBS	45
mq.m7g.4xlarge	16	64	7,5 / 15,0	Production	EBS	90
mq.m7g.8xlarge	32	128	15 gigabits	Production	EBS	175

Type d'instance	vCPU	Mémoire (Gio)	Ligne de base du réseau/ bande passante en rafale (Gbit/s)	Usage recommandé	Stockage	Taille du volume de disque par nœud (Go)
mq.m7g.12xlarge	48	192	22,5 gigabits	Production	EBS	260
mq.m7g.16xlarge	64	256	30 gigabits	Production	EBS	345

Types d'instances pour le déploiement d'une instance unique m7g

Le tableau suivant indique les types d'`mq.m7g.x`instances disponibles pour le déploiement d'une seule instance.

Type d'instance	vCPU	Mémoire (Gio)	Ligne de base du réseau/ bande passante en rafale (Gbit/s)	Usage recommandé	Stockage	Taille du volume de disque par nœud (Go)
mq.m7g.medium	1	4	0,52 / 12,5	Évaluation	EBS	200
mq.m7g.large	2	8	0,937 / 12,5	Production	EBS	200
mq.m7g.xlarge	4	16	1,876/ 12,5	Production	EBS	200

Type d'instance	vCPU	Mémoire (Gio)	Ligne de base du réseau/ bande passante en rafale (Gbit/s)	Usage recommandé	Stockage	Taille du volume de disque par nœud (Go)
mq.m7g.2xlarge	8	32	3,75 / 15,0	Production	EBS	200
mq.m7g.4xlarge	16	64	7,5 / 15,0	Production	EBS	200
mq.m7g.8xlarge	32	128	15 gigabits	Production	EBS	200
mq.m7g.12xlarge	48	192	22,5 gigabits	Production	EBS	200
mq.m7g.16xlarge	64	256	39 gigabits	Production	EBS	200

Types d'instances pour le déploiement **mq.m5** d'une seule instance

Les tableaux suivants présentent les types d'`mq.m5.x`instances disponibles pour le déploiement d'une seule instance.

Type d'instance	vCPU	Mémoire (Gio)	Ligne de base du réseau/ bande passante en rafale (Gbit/s)	Usage recommandé	Stockage	Taille du volume de disque par nœud (Go)
mq.t3.micro	2	1	0,064/5,0	Évaluation	EBS	20
mq.m5.large	2	8	0,75/ 10,0	Production	EBS	200
mq.m5.xlarge	4	16	1,25 / 10,0	Production	EBS	200
mq.m5.2xlarge	8	32	2,5 / 10,0	Production	EBS	200
mq.m5,4 x large	16	64	5,0 / 10,0	Production	EBS	200

Types d'instances pour le déploiement de **`mq.m5`** clusters

Les tableaux suivants présentent les types d'`mq.m5.x`instances disponibles pour le déploiement de clusters

Type d'instance	vCPU	Mémoire (Gio)	Ligne de base du réseau/ bande passante en rafale (Gbit/s)	Usage recommandé	Stockage	Taille du volume de disque par nœud (Go)
mq.m5.large	2	8	0,75/ 10,0	Production	EBS	200
mq.m5.xlarge	4	16	1,25 / 10,0	Production	EBS	200
mq.m5.2xlarge	8	32	2,5 / 10,0	Production	EBS	200
mq.m5,4 x large	16	64	5,0 / 10,0	Production	EBS	200

Directives de dimensionnement d'Amazon MQ pour RabbitMQ

Vous pouvez choisir le type d'instance de courtier le mieux adapté à votre application. Lorsque vous choisissez un type d'instance, tenez compte des facteurs qui affecteront les performances du courtier :

- le nombre de clients et de files d'attente
- le volume de messages envoyés
- messages conservés en mémoire
- messages redondants

`m7g.medium`Les types d'instances de broker plus petits sont recommandés uniquement pour tester les performances des applications. Nous recommandons des types `m7g.large` d'instances de courtier plus grands ou supérieurs ou des niveaux de production de clients et de files d'attente, un débit élevé, des messages en mémoire et des messages redondants.

> **Important**
>
> Vous ne pouvez pas rétrograder un broker d'un type d'`mq.m7g`instance `mq.m5` ou vers un type d'`mq.t3.micro`instance.

Il est important de tester vos courtiers afin de déterminer le type et la taille d'instance appropriés à vos exigences en matière de messagerie de charge de travail.

Utilisez toujours les limites de ressources par défaut du courtier RabbitMQ 4 pour déterminer la taille d'instance appropriée pour votre application conformément aux meilleures pratiques d'Amazon MQ. Ces limites de ressources par défaut sont basées sur les types, le type d'`m7g`instance et les files d'attente de quorum.

- Limites de ressources par défaut pour le déploiement d'une instance unique m7g
- Limites de ressources par défaut pour le déploiement de clusters m7g

Vous pouvez augmenter la valeur de n'importe quelle limite jusqu'aux valeurs maximales définies par type d'instance et mode de déploiement. Cependant, nous vous recommandons vivement de tester les performances du courtier avec les valeurs accrues avant de l'utiliser en production.

- Limites de ressources maximales pour le déploiement d'une instance unique m7g
- Limites de ressources maximales pour le déploiement de clusters m7g
- Limites de ressources maximales pour le déploiement d'une instance unique m5
- Limites de ressources maximales pour le déploiement de clusters m5
- Messages d'erreur

> **Note**
>
> Les courtiers RabbitMQ 3.13 ne sont pas soumis à des limites de ressources par défaut, mais nous vous recommandons d'utiliser les valeurs par défaut suggérées.

Limites de ressources par défaut

Amazon MQ pour RabbitMQ prend en charge la configuration des limites de ressources du courtier à partir de RabbitMQ 4. Lorsque vous créez un courtier, Amazon MQ applique automatiquement des valeurs par défaut à ces limites de ressources. Ces valeurs par défaut servent de garde-fous pour protéger la disponibilité de votre courtier tout en tenant compte des habitudes d'utilisation courantes des clients. Vous pouvez personnaliser le comportement de votre courtier en modifiant les valeurs de configuration des limites afin de mieux répondre à vos exigences spécifiques en matière de charge de travail.

Avant d'apporter des modifications, veuillez noter que :

Important

1. Les modifications de configuration peuvent avoir un impact sur les performances et la disponibilité des courtiers
2. Comprenez l'impact avant de modifier les options de configuration par défaut
3. Testez d'abord les modifications de configuration dans les environnements hors production
4. Surveillez l'état du courtier après avoir appliqué les modifications

Important

Les valeurs par défaut et les plages prises en charge pour ces configurations varient selon la version de RabbitMQ, le type d'instance et le mode de déploiement du courtier.

Important

Remarque : l'association ou la création d'un broker avec des valeurs de configuration situées en dehors de la plage prise en charge entraînera une réponse d'erreur.

Les limites de ressources par défaut appliquées aux courtiers RabbitMQ 4.2 sont

- Limites de ressources par défaut pour le déploiement d'une instance unique m7g
- Limites de ressources par défaut pour le déploiement de clusters m7g

Limites de ressources par défaut

> **Important**
>
> Amazon MQ pour les courtiers RabbitMQ 3, la valeur par défaut est configurée avec la limite de ressources maximale et Amazon MQ ne permet pas de remplacer la configuration de la limite de ressources.

Valeurs par défaut pour les courtiers à instance unique

Type d'instance	Connexions par nœud	Canaux par nœud	Consommateurs par canal	Files d'attente	hôtes virtuels	Pelles	Echanges	Taille du message en octets
mq.m7g.medium	100	500	10	500	10	30	500	524288
mq.m7g.large	1 500	4 500	10	1 000	50	50	1 000	524288
mq.m7g.xlarge	3 000	9 000	10	2 000	100	100	2 000	524288
mq.m7g.2large	6 000	18 000	10	4 000	150	200	4 000	524288
mq.m7g.4large	12 000	36 000	10	8 000	200	400	8 000	524288
mq.m7g.8large	24 000	72 000	10	16,000	250	800	16,000	524288
mq.m7g.1xlarge	36 000	108 000	10	24 000	300	1 200	24 000	524288

Type d'instance	Connexions par nœud	Canaux par nœud	Consommateurs par canal	Files d'attente	hôtes virtuels	Pelles	Echanges	Taille du message en octets
mq.m7g.1xlarge	48 000	144 000	10	32 000	350	1 600	32 000	524288

Valeurs par défaut pour les courtiers en clusters

Type d'instance	Connexions par nœud	Canaux par nœud	Consommateurs par canal	Files d'attente	hôtes virtuels	Pelles	Echanges	Taille du message en octets
mq.m7g.medium	100	300	10	100	10	10	100	524288
mq.m7g.large	500	1 500	10	1 000	50	30	1 000	524288
mq.m7g.xarge	1 000	3000	10	2 000	100	60	2 000	524288
mq.m7g.2large	2000	6 000	10	4 000	150	120	4 000	524288
mq.m7g.4large	4000	12 000	10	8 000	200	240	8 000	524288
mq.m7g.8large	8000	24 000	10	16,000	250	480	16,000	524288
mq.m7g.1xlarge	12 000	36 000	10	24 000	300	720	24000	524288

Type d'instance	Connexions par nœud	Canaux par nœud	Consommateurs par canal	Files d'attente	hôtes virtuels	Pelles	Echanges	Taille du message en octets
mq.m7g.1xlarge	16,000	48 000	10	32 000	350	960	32 000	524288

Limite de ressources maximale d'Amazon MQ pour RabbitMQ

Directives de dimensionnement pour m7g avec files d'attente de quorum pour le déploiement d'une instance unique

Le tableau suivant indique les valeurs limites maximales pour chaque type d'instance pour les courtiers à instance unique.

Type d'instance	Connexions	Canaux	Consommateurs par canal	Files d'attente	Hôtes virtuels	Pelles	Echanges	Taille du message en octets
mq.m7g.medium	300	900	1 000	2 500	10	150	12500	134217728
mq.m7g.large	5 000	15 000	1 000	20 000	1 500	250	100 000	134217728
mq.m7g.xarge	10 000	30 000	1 000	30 000	1 500	500	150 000	134217728
mq.m7g.2large	20 000	60 000	1 000	40 000	1 500	1 000	200 000	134217728
mq.m7g.4large	40 000	120 000	1 000	60 000	1 500	2000	300,000	134217728

Type d'instance	Connexions	Canaux	Consommateurs par canal	Files d'attente	Hôtes virtuels	Pelles	Echanges	Taille du message en octets
mq.m7g.8large	80 000	240 000	1 000	80 000	1 500	4000	400 000	134217728
mq.m7g.12xlarge	120 000	360 000	1 000	100 000	1 500	6 000	500 000	134217728
mq.m7g.16xlarge	160 000	480 000	1 000	120 000	1 500	8 000	600 000	134217728

Directives de dimensionnement pour m7g avec files d'attente de quorum pour le déploiement de clusters

Le tableau suivant indique les valeurs limites maximales pour chaque type d'instance pour les courtiers de clusters.

Type d'instance	Connexions par nœud	Canaux par nœud	Consommateurs par canal	Files d'attente	Hôtes virtuels	Pelles	Echanges	Taille du message en octets
mq.m7g.medium	300	900	1 000	500	10	50	500	134217728
mq.m7g.large	5 000	15 000	1 000	10 000	1 500	150	50 000	134217728
mq.m7g.xlarge	10 000	30 000	1 000	15 000	1 500	300	75 000	134217728

Type d'instance	Connexions par nœud	Canaux par nœud	Consommateurs par canal	Files d'attente	Hôtes virtuels	Pelles	Echanges	Taille du message en octets
mq.m7g.2large	20 000	60 000	1 000	20 000	1 500	600	100 000	134217728
mq.m7g.4large	40 000	120 000	1 000	30 000	1 500	1200	150 000	134217728
mq.m7g.8large	80 000	240 000	1 000	40 000	1 500	2 400	200 000	134217728
mq.m7g.1xlarge	120 000	360 000	1 000	50 000	1 500	3 600	250 000	134217728
mq.m7g.1xlarge	160 000	480 000	1 000	60 000	1 500	4 800	300,000	134217728

Limites de ressources maximales pour le déploiement d'une instance unique M5

Le tableau suivant indique les valeurs limites maximales pour chaque type d'instance pour les courtiers à instance unique.

Type d'instance	Connexions	Canaux	Consommateurs par canal	Files d'attente	Hôtes virtuels	Pelles
m5.large	5 000	15 000	1 000	30 000	1 500	250
m5.xlarge	10 000	30 000	1 000	60 000	1 500	500
m5.2xlarge	20 000	60 000	1 000	120 000	1 500	1 000
m5.4xlarge	40 000	120 000	1 000	240 000	1 000	2 000

Limites de ressources maximales pour le déploiement de clusters m5

Le tableau suivant indique les valeurs limites maximales pour chaque type d'instance pour les courtiers de clusters.

Type d'instance	Files d’attente	Consommateurs par canal	Pelles
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

Les limites de connexion et de canal suivantes sont appliquées par nœud :

Type d'instance	Connexions	Canaux
m5.large	5000	15 000
m5.xlarge	10 000	30 000
m5.2xlarge	20 000	60 000
m5.4xlarge	40 000	120 000

Les valeurs limites exactes pour un courtier de clusters peuvent être inférieures à la valeur indiquée en fonction du nombre de nœuds disponibles et de la manière dont RabbitMQ distribue les ressources entre les nœuds disponibles. Si vous dépassez les valeurs limites, vous pouvez créer une nouvelle connexion à un autre nœud et réessayer, ou vous pouvez augmenter la taille de l'instance pour augmenter les limites maximales

Messages d’erreur

Les messages d'erreur suivants sont renvoyés lorsque les limites sont dépassées. Toutes les valeurs sont basées sur les limites des instances **`m7.large`** uniques.

> **Note**
>
> Les codes d'erreur des messages suivants peuvent changer en fonction de la bibliothèque cliente que vous utilisez.

Connection

```
ConnectionClosedByBroker 500 "NOT_ALLOWED - connection refused: node
connection limit (5000) is reached"
```

Channel

```
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)"
```

Consommateur

```
ConnectionClosedByBroker: (530, 'NOT_ALLOWED - reached maximum (1,000) of
consumers per channel')
```

Taille maximale du message

```
(406, 'PRECONDITION_FAILED - message size 524289 is larger than configured
max size 524288')
```

Échange

```
(406, "PRECONDITION_FAILED - cannot declare exchange 'limit_test_3' in
vhost '/': exchange limit of 10 is reached")
```

> **Note**
>
> Les messages d'erreur suivants utilisent le format de l'API de gestion HTTP.

File d'attente

```
{"error":"bad_request","reason":"cannot declare queue 'my_queue': queue
limit in cluster (10,000) is reached"}]
```

Pelle

```
{"error":"bad_request","reason":"Validation failed\n\ncomponent shovel is
limited to 150 per node\n"}
```

Vhost

```
{"error":"bad_request","reason":"cannot create vhost 'my_vhost': vhost
limit of 1500 is reached"}
```

Valeurs par défaut d'agent Amazon MQ for RabbitMQ

Lorsque vous créez un agent Amazon MQ for RabbitMQ, Amazon MQ applique un ensemble par défaut de politiques d'agent et de limites de vhost pour optimiser les performances de votre agent. Amazon MQ applique des limites de vhost uniquement à la valeur vhost par défaut (`/`). Amazon MQ n'appliquera pas de politiques par défaut aux vhosts nouvellement créés. Nous vous recommandons de conserver ces valeurs par défaut pour tous les agents nouveaux et existants. Toutefois, vous pouvez modifier, remplacer ou supprimer ces valeurs par défaut à tout moment.

Amazon MQ crée différentes politiques de courtage et limites d'hôtes virtuels pour Amazon MQ pour RabbitMQ 3 et RabbitMQ 4. Les différences seront discutées en détail dans les sous-sections suivantes.

Amazon MQ crée des politiques et des limites en fonction du type d'instance et du mode de déploiement de l'agent que vous choisissez lorsque vous créez votre agent. Les politiques par défaut sont nommées en fonction du mode de déploiement, comme suit :

Amazon MQ pour RabbitMQ 3 :

- Instance unique : `AWS-DEFAULT-POLICY-SINGLE-INSTANCE`
- Déploiement de clusters — `AWS-DEFAULT-POLICY-CLUSTER-MULTI-AZ` && `AWS-DEFAULT-QUORUM-QUEUES-POLICY-CLUSTER-MULTI-AZ`

Amazon MQ pour RabbitMQ 4 :

- Instance unique : `AWS-DEFAULT-POLICY-SINGLE-INSTANCE`
- Déploiement de clusters — `AWS-DEFAULT-POLICY-CLUSTER` && `AWS-DEFAULT-QUORUM-QUEUES-POLICY-CLUSTER-MULTI-AZ`

Pour des agents à instance unique, Amazon MQ définit la valeur de priorité de la politique sur `0`. Pour remplacer la valeur de la priorité par défaut, vous pouvez créer vos propres politiques personnalisées avec des valeurs de priorité supérieures. Pour les déploiements en cluster, Amazon MQ définit la valeur de priorité sur 1 pour les valeurs par défaut de l'agent. Pour créer votre propre politique personnalisée pour les clusters, affectez une valeur de priorité supérieure à 1.

> **Note**
>
> Dans les déploiements en clusters, les politiques d'agent `ha-mode` et `ha-sync-mode` sont requises pour la mise en miroir classique et la haute disponibilité (HA). Ces paramètres s'appliquent uniquement à Amazon MQ pour RabbitMQ 3 et ne sont pas configurés pour RabbitMQ 4.
>
> Si vous supprimez la politique par défaut `AWS-DEFAULT-POLICY-CLUSTER-MULTI-AZ`, Amazon MQ utilise la politique `ha-all-AWS-OWNED-DO-NOT-DELETE` avec une valeur de priorité `0`. Cela garantit que les politiques `ha-mode` et `ha-sync-mode` sont toujours en vigueur. Si vous créez votre propre politique personnalisée, Amazon MQ ajoute automatiquement `ha-mode` et `ha-sync-mode` à vos définitions de politique.

Rubriques

Description des politiques et des limites

La liste suivante décrit les politiques et les limites par défaut qu'Amazon MQ applique à un agent nouvellement créé. Les valeurs pour `max-length`, `max-queues` et `max-connections` varient en fonction du type d'instance et du mode de déploiement de votre agent. Ces valeurs sont répertoriées dans la section Valeurs par défaut recommandées.

Paramètres sur les courtiers RabbitMQ 3 et RabbitMQ 4

- **`queue-mode: lazy`** (politique) : active les files d'attente paresseuses. Par défaut, les files d'attente conservent un cache en mémoire de messages, ce qui permet à l'agent de livrer les messages aux consommateurs le plus rapidement possible. Cela peut entraîner un manque de mémoire pour l'agent et déclencher une alarme de mémoire élevée. Les files d'attente paresseuses tentent de déplacer les messages sur le disque dès que possible. Cela signifie que moins de

messages sont conservés en mémoire dans des conditions normales de fonctionnement. En utilisant des files d'attente paresseuses, Amazon MQ for RabbitMQ peut prendre en charge des charges de messagerie beaucoup plus importantes et des files d'attente plus longues. Notez que pour certains cas d'utilisation, les agents avec des files d'attente paresseuses peuvent avoir des performances légèrement plus lentes. Cela est dû au fait que les messages sont déplacés d'un disque à un agent, au lieu de remettre des messages à partir d'un cache en mémoire.

> Modes de déploiement
>
> Instance unique, cluster

- **`max-length: number-of-messages`** (politique) : définit une limite pour le nombre de messages dans une file d'attente. Dans les déploiements en cluster, la limite empêche la synchronisation de file d'attente interrompue dans des cas tels que le redémarrage de l'agent ou à la suite d'une fenêtre de maintenance.

  > Modes de déploiement
  >
  > Cluster

- **`overflow: reject-publish`** (politique) : Applique les files d'attente avec une politique `max-length` pour rejeter les nouveaux messages une fois que le nombre de messages dans la file d'attente atteint la valeur `max-length`. Pour s'assurer que les messages ne sont pas perdus si une file d'attente est dans un état de débordement, les applications client qui publient des messages auprès de l'agent doivent implémenter les confirmations de l'éditeur. Pour plus d'informations sur l'implémentation des confirmations de l'éditeur, consultez Confirmations de l'éditeur sur le site web RabbitMQ.

  > Modes de déploiement
  >
  > Cluster

Paramètres spécifiques à RabbitMQ 3

- **`max-queues: number-of-queues-per-vhost`** (limite vhost) : Définit la limite pour le nombre de files d'attente dans un agent. Similaire à la définition de politique `max-length`, la limitation du nombre de files d'attente dans les déploiements en cluster empêche la synchronisation des

files d'attente interrompues à la suite du redémarrage de l'agent ou des fenêtres de maintenance. La limitation des files d'attente empêche également une utilisation excessive de l'UC pour la maintenance des files d'attente.

> **Modes de déploiement**
>
> Instance unique, cluster

- **`max-connections: number-of-connections-per-vhost`** (limite de vhost) : Définit la limite du nombre de connexions client à l'agent. Limiter le nombre de connexions selon les valeurs recommandées empêche une utilisation excessive de la mémoire de l'agent, ce qui pourrait entraîner l'agent à déclencher une alarme de mémoire élevée et à suspendre les opérations.

  > **Modes de déploiement**
  >
  > Instance unique, cluster

Valeurs par défaut recommandées

> **Important**
>
> `max-queue`set ne `max-connections` sont appliqués qu'à Amazon MQ pour RabbitMQ 3.

> **Note**
>
> Les limites par défaut `max-length` et `max-queue` sont testées et évaluées en fonction d'une taille moyenne de message de 5 Ko. Si vos messages sont nettement supérieurs à 5 Ko, vous devrez ajuster et réduire les limites `max-length` et `max-queue`.

Le tableau suivant répertorie les valeurs limites par défaut pour un agent nouvellement créé. Amazon MQ applique ces valeurs en fonction du type d'instance et du mode de déploiement de l'agent.

Type d'instance	Mode de déploiement	**max-length**	**max-queues**	**max-connections**
mq.m7g.medium	Instance unique	N/A	2 500	100
	Cluster	500 000	100	100
mq.m7g.large	Instance unique	N/A	20 000	5 000
	Cluster	8 000 000	10 000	5 000
mq.m7g.xlarge	Instance unique	N/A	30 000	10 000
	Cluster	9 000 000	15 000	10 000
mq.m7g.2xlarge	Instance unique	N/A	40 000	20 000
	Cluster	10 000 000	40 000	20 000
mq.m7g.4xlarge	Instance unique	N/A	60 000	40 000
	Cluster	12 000 000	30 000	40 000
mq.m7g.8xlarge	Instance unique	N/A	80 000	80 000
	Cluster	20 000 000	40 000	80 000
mq.m7g.12xlarge	Instance unique	N/A	100 000	120 000
	Cluster	30 000 000	20 000	120 000
mq.m7g.16xlarge	Instance unique	N/A	120 000	160 000
	Cluster	40 000 000	50 000	160 000

Type d'instance	Mode de déploiement	**max-length**	**max-queues**	**max-connections**
t3.micro	Instance unique	N/A	500	500

Type d'instance	Mode de déploiement	`max-length`	`max-queues`	`max-connections`
m5.large	Instance unique	N/A	20 000	4 000
m5.large	Cluster	8 000 000	10 000	15 000
m5.xlarge	Instance unique	N/A	30 000	8 000
m5.xlarge	Cluster	9 000 000	10 000	20 000
m5.2xlarge	Instance unique	N/A	60 000	15 000
m5.2xlarge	Cluster	10 000 000	10 000	40 000
m5.4xlarge	Instance unique	N/A	150 000	30 000
m5.4xlarge	Cluster	12 000 000	10 000	100 000

Configuration d'un courtier RabbitMQ

Une configuration contient tous les paramètres de votre broker RabbitMQ au format Cuttlefish. Vous pouvez créer une configuration avant de créer des agents. Vous pouvez ensuite appliquer la configuration à un ou plusieurs agents.

Attributes

La configuration d'un agent a plusieurs attributs, par exemple :

- Un nom (MyConfiguration)
- Un identifiant (c-1234a5b6-78cd-901e-2fgh-3i45j6k178l9)
- Un nom de ressource Amazon (ARN) (arn:aws:mq:us-east-2:123456789012:configuration:c-1234a5b678cd-901e-2fgh-3i45j6k178l9)

Pour obtenir la liste complète des attributs de configuration, consultez ce qui suit dans la référence d'API REST Amazon MQ :

- ID d'opération REST : Configuration

- ID d'opération REST : Configurations

Pour obtenir la liste complète des attributs des révisions de configuration, consultez les sections suivantes :

- ID d'opération REST : Révision de configuration
- ID d'opération REST : Révisions de configuration

Rubriques

- Création et application de configurations de broker RabbitMQ
- Modifier une révision de configuration Amazon MQ pour RabbitMQ
- Valeurs configurables pour RabbitMQ sur Amazon MQ
- Support de l'ARN dans la configuration de RabbitMQ

Création et application de configurations d'agent RabbitMQ

Une configuration contient tous les paramètres de votre agent RabbitMQ au format Cuttlefish. Vous pouvez créer une configuration avant de créer des agents. Vous pouvez ensuite appliquer la configuration à un ou plusieurs agents.

Les exemples suivants montrent comment créer une configuration d'agent RabbitMQ et comment l'appliquer à l'aide de la AWS Management Console.

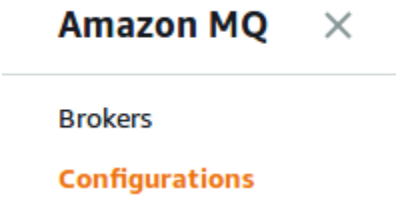
> **Important**
>
> Vous ne pouvez supprimer une configuration qu'à l'aide de l'`DeleteConfiguration`API. Pour plus d'informations, consultez Configurations dans le manuel Amazon MQ API Reference.

Création d'une nouvelle configuration

Pour appliquer une configuration à votre broker, vous devez d'abord créer la configuration.

1. Connectez-vous à la console Amazon MQ.

2. Sur la gauche, développez le volet de navigation et choisissez Configurations.



3. Sur la page Configurations, choisissez Create configuration (Créer une configuration).
4. Sur la page Create configuration (Créer une configuration), dans la section Details (Détails), saisissez le Configuration name (Nom de configuration) (par exemple, `MyConfiguration`) et sélectionnez une version de Broker engine (Moteur d'agent).

   Pour en savoir plus sur les versions du moteur RabbitMQ prises en charge par Amazon MQ for RabbitMQ, consultez the section called “Gestion des versions”.
5. Choisissez Créer une configuration.

Créer une révision de configuration

Après avoir créé une configuration, vous devez la modifier à l'aide d'une révision de configuration.

1. Dans la liste des configurations, choisissez ***`MyConfiguration`***.

   > **Note**
   >
   > La première révision de configuration est toujours créée lorsqu'Amazon MQ crée la configuration.

   Sur la ***`MyConfiguration`***page, le type et la version du moteur de courtage utilisés par votre nouvelle révision de configuration (par exemple, RabbitMQ 3.xx.xx) sont affichés.
2. Dans l'onglet Détails de configuration figurent le numéro de révision de configuration, la description et la configuration d'agent au format Cuttlefish.

   > **Note**
   >
   > La modification de la configuration actuelle crée une nouvelle révision de configuration.
3. Choisissez Modifier la configuration et apportez les modifications à la configuration Cuttlefish.

4. Choisissez Enregistrer.

   La boîte de dialogue Save revision (Enregistrer la révision) s'affiche.

5. (Facultatif) Type `A description of the changes in this revision`.

6. Choisissez Save (Enregistrer).

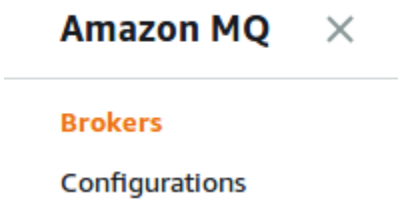
   La nouvelle révision de configuration est enregistrée.

> **Important**
>
> Apporter des modifications à une configuration n'applique pas immédiatement les modifications à l'agent. Pour appliquer vos modifications, vous devez attendre la fenêtre de maintenance suivante ou redémarrer l'agent.
> Actuellement, vous ne pouvez pas supprimer une configuration.

Appliquer une révision de configuration à votre agent

Après avoir créé la révision de configuration, vous pouvez appliquer la révision de configuration à votre courtier.

1. Sur la gauche, développez le volet de navigation et choisissez Brokers (Agents).



2. Dans la liste des courtiers, sélectionnez votre courtier (par exemple MyBroker), puis choisissez Modifier.

3. Sur la *`MyBroker`* page Modifier, dans la section Configuration, sélectionnez une configuration et une révision, puis choisissez Planifier les modifications.

4. Dans la section Schedule broker modifications (Planifier les modifications de l'agent), choisissez si les modifications doivent être appliquées During the next scheduled maintenance window (Au cours de la prochaine fenêtre de maintenance) ou Immediately (immédiatement).

> **Important**
>
> Les courtiers à instance unique sont hors ligne lors du redémarrage. Pour les courtiers en clusters, un seul nœud est en panne à la fois lorsque le courtier redémarre.

5. Cliquez sur Appliquer.

   Votre révision de configuration est appliquée à votre agent à l'heure spécifiée.

Modifier une révision de configuration Amazon MQ pour RabbitMQ

Les instructions suivantes décrivent comment modifier une révision de configuration pour votre courtier.

1. Connectez-vous à la console Amazon MQ.
2. Dans la liste des courtiers, sélectionnez votre courtier (par exemple MyBroker), puis choisissez Modifier.
3. Sur la ***MyBroker***page, choisissez Modifier.
4. Sur la *MyBroker* page Modifier, dans la section Configuration, sélectionnez une configuration et une révision, puis choisissez Modifier.

   > **Note**
   >
   > Sauf si vous sélectionnez une configuration lorsque vous créez un agent, la première révision de configuration est toujours créée lorsqu'Amazon MQ crée l'agent.

   Sur la ***MyBroker***page, le type et la version du moteur de courtage utilisés par la configuration (par exemple, RabbitMQ 3.xx.xx) sont affichés.
5. Dans l'onglet Détails de configuration figurent le numéro de révision de configuration, la description et la configuration d'agent au format Cuttlefish.

   > **Note**
   >
   > La modification de la configuration actuelle crée une nouvelle révision de configuration.

6. Choisissez Modifier la configuration et apportez les modifications à la configuration Cuttlefish.
7. Choisissez Enregistrer.

   La boîte de dialogue Save revision (Enregistrer la révision) s'affiche.
8. (Facultatif) Type `A description of the changes in this revision`.
9. Choisissez Save (Enregistrer).

   La nouvelle révision de configuration est enregistrée.

> **Important**
>
> Apporter des modifications à une configuration n'applique pas immédiatement les modifications à l'agent. Pour appliquer vos modifications, vous devez attendre la fenêtre de maintenance suivante ou redémarrer l'agent.
> Actuellement, vous ne pouvez pas supprimer une configuration.

Valeurs configurables

Vous pouvez définir la valeur des options de configuration de courtier suivantes en modifiant le fichier de configuration de courtier dans le AWS Management Console.

Outre les valeurs décrites dans le tableau suivant, Amazon MQ prend en charge des options de configuration de courtier supplémentaires liées à l'authentification et à l'autorisation ainsi qu'aux limites de ressources. Pour plus d'informations sur ces options de configuration, voir

- OAuth Configuration 2.0
- Configuration LDAP
- Configuration du protocole HTTP
- Configuration du protocole SSL
- Configuration de MTLS
- Prise en charge de l'ARN
- Limites des ressources
- Configuration SSL du client AMQP

Configuration	Valeur par défaut	Valeur recommandée	Valeurs	Versions applicables	Description
consumer_timeout	1800000 ms (30 minutes)	1800000 ms (30 minutes)	0 à 2 147 483 647 ms. Amazon MQ prend également en charge la valeur 0, qui signifie « infini ».	Toutes les versions	Un délai d'attente pour l'accusé de réception pour détecter les cas où les consommateurs n'emballent pas les livraisons.
battement de coeur	60 secondes	60 secondes	60 à 3600 secondes	Toutes les versions	Définit le délai avant qu'une connexion ne soit considérée comme indisponible par RabbitMQ.
management.restrictions.operator_policy_changes.disabled	true	true	true, false	Toutes les versions	Désactive la modification des politiques de l'opérateur. Si vous apportez cette modification, nous vous encourageons vivement

Configuration	Valeur par défaut	Valeur recommandée	Valeurs	Versions applicables	Description
					à inclure les propriétés de haute disponibilité dans vos propres politiques d'opérateur.
quorum_queue.property_equivalence.relaxed_checks_on_redeclaration	true	true	true, false	Toutes les versions	Lorsque cette valeur est définie sur TRUE, votre application évite une exception de canal lors de la redéclaration d'une file d'attente de quorum.
secure.management.http.headers.enabled	true	true	true, false	Toutes les versions	Active les en-têtes de sécurité HTTP non modifiables.

Configuration de l'accusé de réception pour le consommateur

Vous pouvez configurer consumer_timeout pour détecter les cas où les consommateurs n'emballent pas leurs livraisons. Si le consommateur n'envoie pas d'accusé de réception dans le délai imparti, le canal sera fermé. Par exemple, si vous utilisez la valeur par défaut de 1 800 000 millisecondes, si

le consommateur n'envoie pas d'accusé de réception dans un délai de 1 800 000 millisecondes, le canal sera fermé. Amazon MQ prend également en charge la valeur 0, qui signifie « infini ».

Configuration du rythme cardiaque

Vous pouvez configurer un délai d'expiration du rythme cardiaque pour savoir quand les connexions sont interrompues ou ont échoué. La valeur du rythme cardiaque définit le délai avant qu'une connexion ne soit considérée comme interrompue.

Configuration des politiques de l'opérateur

La politique d'opérateur par défaut de chaque hôte virtuel présente les propriétés de haute disponibilité recommandées suivantes :

```
{
"name": "default_operator_policy_AWS_managed",
"pattern": ".*",
"apply-to": "all",
"priority": 0,
"definition": {
"ha-mode": "all",
"ha-sync-mode": "automatic"
}
}
```

Les modifications apportées aux politiques de l'opérateur via l'API de gestion AWS Management Console or ne sont pas disponibles par défaut. Vous pouvez activer les modifications en ajoutant la ligne suivante à la configuration d'agent :

```
management.restrictions.operator_policy_changes.disabled=false
```

Si vous apportez cette modification, nous vous encourageons vivement à inclure les propriétés de haute disponibiliété dans vos propres politiques d'opérateur.

Configuration de contrôles simplifiés lors de la déclaration des files d'attente

Si vous avez migré vos files d'attente classiques vers des files d'attente de quorum mais que vous n'avez pas mis à jour votre code client, vous pouvez éviter une exception de canal lorsque vous redéclarez une file d'attente de quorum en configurant quorum_queue.property_equivalence.relaxed_checks_on_redeclaration sur true.

Configuration des en-têtes de sécurité HTTP

La configuration secure.management.http.headers.enabled active les en-têtes de sécurité HTTP suivants :

- X-Content-Type-Options : nosniff : empêche les navigateurs de détecter du contenu, algorithmes utilisés pour déduire le format de fichier des sites Web.
- X-Frame-Options : DENY : empêche les autres d'intégrer le plugin de gestion dans un cadre de leur propre site Web pour tromper les autres
- Strict-Transport-Security : max-age=47304000 ; includeSubDomains : oblige les navigateurs à utiliser le protocole HTTPS lors de connexions ultérieures au site Web et à ses sous-domaines pendant une longue période (1,5 an).

Les courtiers Amazon MQ pour RabbitMQ créés sur les versions 3.10 et supérieures auront la valeur true par défaut pour secure.management.http.headers.enabled. Vous pouvez activer ces en-têtes de sécurité HTTP en attribuant à secure.management.http.headers.enabled la valeur true. Si vous souhaitez désactiver ces en-têtes de sécurité HTTP, définissez secure.management.http.headers.enabled sur false.

Configuration de l'authentification et de l'autorisation OAuth 2.0

Pour plus d'informations sur les options de configuration OAuth 2.0 et sur la configuration de l'authentification OAuth 2.0 pour vos courtiers, consultez les sections Configurations OAuth 2.0 prises en charge et Utilisation de l'authentification et de l'autorisation OAuth 2.0.

Configuration de l'authentification et de l'autorisation LDAP

Pour plus d'informations sur les options de configuration LDAP et sur la configuration de l'authentification LDAP pour vos courtiers, consultez les sections Configurations LDAP prises en charge et. Utilisation de l'authentification et de l'autorisation LDAP

Configuration de l'authentification et de l'autorisation HTTP

Pour plus d'informations sur les valeurs de configuration de l'authentification HTTP et sur la configuration de l'authentification HTTP pour vos courtiers, voir Authentification et autorisation HTTP etUtilisation de l'authentification et de l'autorisation HTTP.

> **Note**
>
> Le plug-in d'authentification HTTP est uniquement disponible pour Amazon MQ pour RabbitMQ version 4 et supérieure.

Configuration de l'authentification par certificat SSL

Pour plus d'informations sur les valeurs de configuration de l'authentification par certificat SSL et sur la configuration de l'authentification par certificat SSL pour vos courtiers, voir Authentification par certificat SSL etUtilisation de l'authentification par certificat SSL.

> **Note**
>
> Le plug-in d'authentification par certificat SSL est uniquement disponible pour Amazon MQ pour RabbitMQ version 4 et supérieure.

Configuration des MTL

Amazon MQ pour RabbitMQ prend en charge le protocole TLS mutuel (MTL) pour des connexions sécurisées à divers points de terminaison et services externes. Le protocole mTLS améliore la sécurité en obligeant le client et le serveur à s'authentifier à l'aide de certificats.

> **Note**
>
> L'utilisation d'autorités de certification privées pour les MTL n'est disponible que pour Amazon MQ pour RabbitMQ version 4 et supérieure.

> **Important**
>
> Amazon MQ pour RabbitMQ impose l'utilisation de fichiers de certificat et de clé privée AWS ARNs . Voir le support de l'ARN dans la configuration de RabbitMQ pour plus de détails.

Sur cette page

- Point de terminaison AMQP
- Plugin de gestion RabbitMQ
- Plug-in RabbitMQ 2.0 OAuth
- Plug-in d'authentification HTTP RabbitMQ
- Plug-in LDAP RabbitMQ
- Connexions client AMQP

Point de terminaison AMQP

Configurez les MTL pour les connexions client au point de terminaison AMQP. Ceci est utilisé avec l'authentification par certificat SSL. Pour les configurations prises en charge, consultezAuthentification par certificat SSL.

Plugin de gestion RabbitMQ

Configurez les MTL pour les connexions à l'interface de gestion RabbitMQ.

> **Note**
>
> Les MTL stricts ne sont pas pris en charge pour l'API de gestion.

Configurations prises en charge

`aws.arns.management.ssl.cacertfile`

Fichier d'autorité de certification pour valider les certificats clients se connectant à l'interface de gestion.

`management.ssl.verify`

Mode de vérification par les pairs. Valeurs prises en charge :`verify_none`, `verify_peer`

`management.ssl.depth`

Profondeur maximale de la chaîne de certificats pour la vérification.

`management.ssl.hostname_verification`

Mode de vérification du nom d'hôte. Valeurs prises en charge :`wildcard`, none

Options SSL non prises en charge

Les valeurs de configuration SSL suivantes ne sont pas prises en charge :

Voir la liste complète

- `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`

Plug-in RabbitMQ 2.0 OAuth

Configurez les MTL pour les connexions entre Amazon MQ et OAuth le fournisseur d'identité 2.0. Pour les configurations prises en charge, consultezOAuth Authentification et autorisation 2.0.

Plug-in d'authentification HTTP RabbitMQ

Configurez les MTL pour les connexions entre Amazon MQ et le serveur d'authentification HTTP. Pour les configurations prises en charge, consultezAuthentification et autorisation HTTP.

Plug-in LDAP RabbitMQ

Configurez les MTL pour les connexions entre Amazon MQ et le serveur LDAP. Pour les configurations prises en charge, consultezAuthentification et autorisation LDAP.

Connexions client AMQP

Configurez la vérification par les pairs TLS pour les connexions client AMQP utilisées par Federation et Shovel. Pour plus d'informations, consultez la section Configuration SSL du client AMQP.

> **Important**
>
> Amazon MQ ne prend actuellement pas en charge la configuration de certificats clients pour les connexions client AMQP. Par conséquent, Federation et Shovel ne peuvent pas se connecter aux courtiers compatibles MTLS qui nécessitent une authentification par certificat client.

Configuration des limites de ressources

Amazon MQ pour RabbitMQ prend en charge la configuration des limites de ressources des courtiers à partir de RabbitMQ 4. Lorsque vous créez un courtier, Amazon MQ applique automatiquement des valeurs par défaut à ces limites de ressources. Ces valeurs par défaut servent de garde-fous pour protéger la disponibilité de votre courtier tout en tenant compte des habitudes d'utilisation courantes des clients. Vous pouvez personnaliser le comportement de votre courtier en modifiant les valeurs de configuration des limites afin de mieux répondre à vos exigences spécifiques en matière de charge de travail. Pour plus de détails sur les valeurs par défaut et maximales autorisées, consultezthe section called “Directives de dimensionnement”.

Noms des ressources et clés de configuration

Nom de la ressource	Clé de configuration
Connexion	`connection_max`
Channel	`channel_max_per_node`
File d’attente	`cluster_queue_limit`

Nom de la ressource	Clé de configuration
Vhost	`vhost_max`
Pelle	`runtime_parameters.limits.shovel`
Exchange	`cluster_exchange_limit`
Consommateur par canal	`consumer_max_per_channel`
Taille maximale du message	`max_message_size`

Comment contourner les limites de ressources

Vous pouvez contourner les limites de ressources à l'aide de l'API Amazon MQ et de la console Amazon MQ.

L'exemple suivant montre comment remplacer la limite par défaut du nombre de files d'attente à l'aide de AWS CLI :

```
aws mq update-configuration --configuration-id <config-id> --data "$(echo
 "cluster_queue_limit=500" | base64 --wrap=0)"
```

Un appel réussi crée une révision de configuration. Vous devez associer la configuration à votre courtier RabbitMQ et redémarrer le courtier pour appliquer la dérogation. Pour plus de détails, voir RabbitMQ Broker Configurations

Erreurs de contournement des limites de ressources

L'association ou la création d'un broker avec des valeurs de configuration situées en dehors de la plage prise en charge entraîne une réponse d'erreur similaire à la suivante :

```
Configuration Revision N for configuration:cluster_queue_limit has limit: of value:
 100000000 larger than maximum allowed limit:5000
```

Support de l'ARN dans la configuration de RabbitMQ

Amazon MQ pour RabbitMQ prend en charge AWS ARNs les valeurs de certains paramètres de configuration de RabbitMQ. Ceci est activé par le plugin communautaire RabbitMQ rabbitmq-aws. Ce plugin est développé et maintenu par Amazon MQ et peut également être utilisé par des courtiers RabbitMQ auto-hébergés qui ne sont pas gérés par Amazon MQ.

⚠ Considérations importantes

- Les valeurs ARN résolues récupérées par le plugin aws sont transmises directement au processus RabbitMQ lors de l'exécution. Ils ne sont pas stockés ailleurs sur le nœud RabbitMQ.
- Amazon MQ pour RabbitMQ nécessite un rôle IAM qui peut être assumé par Amazon MQ pour accéder au configuré. ARNs Ceci est configuré par réglage`aws.arns.assume_role_arn`.
- Les utilisateurs qui appellent CreateBroker ou UpdateBroker APIs dont la configuration de courtier inclut un rôle IAM doivent disposer de l'`iam:PassRole`autorisation pour ce rôle.
- Le rôle IAM doit exister sur le même AWS compte que le courtier RabbitMQ. Tous ARNs les éléments de la configuration doivent être présents dans la même AWS région que le broker RabbitMQ.
- Amazon MQ ajoute des clés conditionnelles globales IAM `aws:SourceAccount` et `aws:SourceArn` lorsqu'il assume le rôle IAM. Ces valeurs doivent être utilisées dans la politique IAM associée au rôle pour éviter toute confusion dans la protection des adjoints.

Sur cette page

Clés prises en charge

Rôle IAM requis

`aws.arns.assume_role_arn`

ARN du rôle IAM qu'Amazon MQ suppose pour accéder AWS à d'autres ressources. Obligatoire lorsqu'une autre configuration d'ARN est utilisée.

Point de terminaison AMQP

Clé de configuration	Description
`aws.arns.ssl_options.cacertfile`	Fichier d'autorité de certification pour les connexions SSL/TLS client. Amazon MQ nécessite l'utilisation d'Amazon S3 ou le stockage du certificat.

Plugin de gestion RabbitMQ

Clé de configuration	Description
`aws.arns.management.ssl.cacertfile`	Fichier d'autorité de certification pour les SSL/TLS connexions à l'interface de gestion. Amazon MQ nécessite l'utilisation d'Amazon S3 ou le stockage du certificat.

Plug-in RabbitMQ 2.0 OAuth

Clé de configuration	Description
`aws.arns.auth_oauth2.https.cacertfile`	Fichier d'autorité de certification pour les connexions HTTPS OAuth 2.0. Amazon MQ nécessite l'utilisation d'Amazon S3 ou le stockage du certificat.

Plug-in d'authentification HTTP RabbitMQ

Clé de configuration	Description
`aws.arns.auth_http.ssl_options.cacertfile`	Fichier d'autorité de certification pour les SSL/TLS connexions d'authentification HTTP. Amazon MQ nécessite l'utilisation d'Amazon S3 ou le stockage du certificat.
`aws.arns.auth_http.ssl_options.certfile`	Fichier de certificat pour les connexions TLS mutuelles entre Amazon MQ et le serveur d'authentification HTTP. Amazon MQ nécessite l'utilisation d'Amazon S3 ou le stockage du certificat.
`aws.arns.auth_http.ssl_options.keyfile`	Fichier de clé privée pour les connexions TLS mutuelles entre Amazon MQ et le serveur d'authentification HTTP. Amazon MQ doit être utilisé AWS Secrets Manager pour stocker la clé privée.

Plug-in LDAP RabbitMQ

Clé de configuration	Description
`aws.arns.auth_ldap.ssl_options.cacertfile`	Fichier d'autorité de certification pour les SSL/TLS connexions LDAP. Amazon MQ nécessite l'utilisation d'Amazon S3 ou le stockage du certificat.
`aws.arns.auth_ldap.ssl_options.certfile`	Fichier de certificat pour les connexions TLS mutuelles entre Amazon MQ et le serveur LDAP. Amazon MQ nécessite l'utilisation d'Amazon S3 ou le stockage du certificat.
`aws.arns.auth_ldap.ssl_options.keyfile`	Fichier de clé privée pour les connexions TLS mutuelles entre Amazon MQ et le serveur LDAP. Amazon MQ doit être utilisé AWS Secrets Manager pour stocker la clé privée.
`aws.arns.auth_ldap.dn_lookup_bind.password`	Mot de passe pour la liaison de recherche du DN LDAP. Amazon MQ nécessite l'utilisation AWS Secrets Manager pour stocker le mot de passe sous forme de valeur en texte brut.

Clé de configuration	Description
`aws.arns.auth_ldap.other_bind.password`	Mot de passe pour l'autre liaison LDAP. Amazon MQ nécessite l'utilisation AWS Secrets Manager pour stocker le mot de passe sous forme de valeur en texte brut.

Exemples de politiques IAM

Pour des exemples de politique IAM, notamment des documents de politique d'acceptation des rôles et des documents de politique de rôle, consultez l'exemple d'implémentation du CDK.

Consultez Utilisation de l'authentification et de l'autorisation LDAP les étapes de configuration AWS Secrets Manager et les ressources Amazon S3.

Validation d'accès

Pour résoudre les scénarios dans lesquels les valeurs ARN ne peuvent pas être récupérées, le plugin aws prend en charge un point de terminaison d'API de gestion RabbitMQ qui peut être appelé pour vérifier si Amazon MQ est en mesure d'assumer correctement le rôle et de le résoudre. AWS ARNs Cela évite d'avoir à mettre à jour la configuration du courtier, à mettre à jour le courtier avec la nouvelle révision de configuration et à redémarrer le courtier pour tester les modifications de configuration.

> **Note**
>
> L'utilisation de cette API nécessite un utilisateur administrateur RabbitMQ existant. Amazon MQ recommande de créer des courtiers de test avec un utilisateur interne en plus des autres méthodes d'accès. Consultez la section Activation à la fois de l'authentification OAuth 2.0 et de l'authentification simple (interne). Cet utilisateur peut ensuite être utilisé pour accéder à l'API de validation.

> **Note**
>
> Bien que le plugin aws prenne en charge le transfert d'un nouveau rôle en tant qu'entrée à l'API de validation, ce paramètre n'est pas pris en charge par Amazon MQ. Le rôle IAM utilisé pour la validation doit correspondre à la valeur de `aws.arns.assume_role_arn` dans la configuration du broker.

États de quarantaine des courtiers associés

Pour plus d'informations sur les états de quarantaine des courtiers liés aux problèmes liés à la prise en charge de l'ARN, voir :

- RABBITMQ_INVALID_ASSUMEROLE
- RABBITMQ_INVALID_ARN_LDAP
- RABBITMQ_ARN_INVALIDE

Exemple de scénario

- `b-f0fc695e-2f9c-486b-845a-988023a3e55b`Le courtier a été configuré pour utiliser le rôle IAM `<role>` pour accéder au secret AWS Secrets Manager `<arn>`
- Si le rôle fourni à Amazon MQ ne dispose pas d'une autorisation de lecture sur le AWS Secrets Manager secret, l'erreur suivante sera affichée dans les journaux de RabbitMQ :

  ```
  [error] <0.254.0> aws_arn_config: {handle_assume_role,{error,{assume_role_failed,"AWS
   service is unavailable"}}}
  ```

  De plus, le courtier entrera dans l'état de `INVALID_ASSUMEROLE` quarantaine. Pour plus d'informations, consultez INVALID_ASSUMEROLE.
- Les tentatives d'authentification LDAP échoueront avec l'erreur suivante :

  ```
  [error] <0.254.0> LDAP bind failed: invalid_credentials
  ```
- Corrigez le rôle IAM avec les autorisations appropriées
- Appelez le point de validation pour vérifier si RabbitMQ est désormais en mesure d'accéder au secret :

  ```
  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 '.'
  ```

Configuration SSL du client AMQP

Federation et Shovel utilisent l'AMQP pour communiquer entre les courtiers en amont et en aval. Par défaut, la vérification par les pairs TLS est activée pour les clients AMQP dans Amazon MQ pour RabbitMQ 4. Avec ce paramètre, les clients AMQP de fédération et de pelle exécutés sur des courtiers Amazon MQ effectueront une vérification par leurs pairs lors de l'établissement de connexions avec le courtier en amont.

Les clients AMQP s'exécutant sur les courtiers Amazon MQ prennent en charge les mêmes autorités de certification que Mozilla. Si vous n'utilisez pas ACM, utilisez un certificat émis par une autorité de certification figurant dans la liste des certificats d'autorité de certification inclus par Mozilla. Si votre courtier local utilise des certificats provenant d'autres autorités de certification, la vérification SSL échouera. Vous pouvez désactiver la vérification par les pairs TLS pour ces cas d'utilisation.

Important

Amazon MQ ne prend actuellement pas en charge la configuration de certificats clients pour les connexions client AMQP. Par conséquent, Federation et Shovel ne peuvent pas se connecter aux courtiers compatibles MTLS qui nécessitent une authentification par certificat client.

Important

Sur Amazon MQ pour RabbitMQ 3, les propriétés SSL des clients AMQP sont configurées avec les valeurs par défaut de RabbitMQ (verify_none). Amazon MQ pour RabbitMQ 3 ne prend pas en charge le remplacement de ces valeurs par défaut.

Note

Avec le `verify_peer` paramètre par défaut, vous pouvez établir des connexions de fédération et de pelle entre deux courtiers Amazon MQ, mais cela ne permet pas d'établir la connexion entre le courtier Amazon MQ et des courtiers privés ou des courtiers sur site qui fonctionnent avec des certificats autres qu'Amazon MQ CA. Pour vous connecter à des courtiers privés ou locaux, vous devez désactiver la vérification par les pairs sur le courtier Amazon MQ en aval.

Clé de configuration SSL du client AMQP

Configuration	Clé de configuration	Valeurs prises en charge
Vérification par les pairs SSL du client AMQP	`amqp_client.ssl_options.verify`	`verify_none` , `verify_peer`

Comment annuler la vérification SSL par les pairs du client AMQP

Vous pouvez annuler la vérification SSL par les pairs du client AMQP à l'aide de l'API Amazon MQ et de la console Amazon MQ sur les courtiers RabbitMQ 4.

L'exemple suivant montre comment annuler la vérification par les pairs SSL du client AMQP à l'aide de : AWS CLI

```
aws mq update-configuration --configuration-id <config-id> --data "$(echo
 "amqp_client.ssl_options.verify=verify_none" | base64 --wrap=0)"
```

Un appel réussi crée une révision de configuration. Vous devez associer la configuration à votre courtier RabbitMQ et redémarrer le courtier pour appliquer la dérogation. Pour plus de détails, voir Creating and applying broker configurations

> **Important**
>
> Lors de l'utilisation`verify_none`, le cryptage SSL est toujours actif, mais l'identité de l'homologue n'est pas vérifiée. Utilisez ce paramètre uniquement lorsque cela est nécessaire et assurez-vous que le chemin réseau menant au courtier de destination est fiable.

Authentification et autorisation Amazon MQ pour RabbitMQ

Amazon MQ pour RabbitMQ prend en charge les méthodes d'authentification et d'autorisation suivantes :

Authentification et autorisation simples

Dans cette méthode, les utilisateurs du courtier sont stockés en interne dans le courtier RabbitMQ et gérés via la console Web ou l'API de gestion. Les autorisations pour les hôtes virtuels, les échanges, les files d'attente et les sujets sont configurées directement dans RabbitMQ. Il s'agit de la méthode par défaut. Pour plus d'informations, consultez Authentification et autorisation simples.

OAuth Authentification et autorisation 2.0

Dans cette méthode, les utilisateurs du broker et leurs autorisations sont gérés par un fournisseur d'identité OAuth 2.0 externe (IdP). L'authentification des utilisateurs et les autorisations de ressources pour les hôtes virtuels, les échanges, les files d'attente et les sujets sont centralisées via le système de périmètre du fournisseur OAuth 2.0. Cela simplifie la gestion des utilisateurs et permet l'intégration aux systèmes d'identité existants. Pour plus d'informations, voir Authentification et autorisation OAuth 2.0.

Authentification et autorisation IAM

Dans cette méthode, les utilisateurs du broker s'authentifient à l'aide des informations d'identification AWS IAM via la fédération sortante IAM. Les informations d'identification IAM sont utilisées pour obtenir des jetons JWT auprès du AWS Security Token Service (STS), et ces jetons JWT servent de jetons OAuth 2.0 pour l'authentification. Cette méthode s'appuie sur le support OAuth 2.0 existant dans Amazon MQ pour RabbitMQ, qui agit en tant que fournisseur d' AWS identité 2.0. OAuth L'authentification des utilisateurs est gérée par AWS IAM, tandis que les autorisations de ressources pour les hôtes virtuels, les échanges, les files d'attente et les sujets sont gérées via des politiques IAM et des alias de portée configurés dans RabbitMQ. Pour plus d'informations, consultez Authentification et autorisation IAM.

Authentification et autorisation LDAP

Dans cette méthode, les utilisateurs du broker et leurs autorisations sont gérés par un service d'annuaire LDAP externe. L'authentification des utilisateurs et les autorisations de ressources sont centralisées via le serveur LDAP, ce qui permet aux utilisateurs d'accéder à RabbitMQ en utilisant leurs informations d'identification de service d'annuaire existantes. Pour plus d'informations, consultez Authentification et autorisation LDAP.

Authentification et autorisation HTTP

Dans cette méthode, les utilisateurs du broker et leurs autorisations sont gérés par un serveur HTTP externe. L'authentification des utilisateurs et les autorisations de ressources sont centralisées via le serveur HTTP, ce qui permet aux utilisateurs d'accéder à RabbitMQ en utilisant leur propre fournisseur d'authentification et d'autorisation. Pour plus d'informations sur cette méthode, consultez Authentification et autorisation HTTP.

Authentification par certificat SSL

Amazon MQ prend en charge le protocole TLS mutuel (MTL) pour les courtiers RabbitMQ. Le plugin d'authentification SSL utilise des certificats clients issus de connexions mTLS pour authentifier les utilisateurs. Dans cette méthode, les utilisateurs du broker sont authentifiés à l'aide de certificats clients X.509 au lieu de leur nom d'utilisateur et de leur mot de passe. Le certificat du client est validé auprès d'une autorité de certification (CA) fiable, et le nom d'utilisateur est extrait d'un champ du certificat, tel que le nom commun (CN) ou le nom alternatif du sujet (SAN). Cette méthode fournit une authentification forte sans transmettre d'informations d'identification sur le réseau. Pour plus d'informations, consultez la section Authentification par certificat SSL.

> **Note**
>
> RabbitMQ prend en charge plusieurs méthodes d'authentification et d'autorisation à utiliser simultanément. Par exemple, vous pouvez activer à la fois l'authentification OAuth 2.0 et l'authentification simple (interne). Pour plus d'informations, consultez la section du didacticiel OAuth 2.0 sur l'activation à la fois de l'authentification OAuth 2.0 et de l'authentification simple (interne) et la documentation sur le contrôle d'accès RabbitMQ.
>
> Amazon MQ recommande de créer un utilisateur interne lors du test des configurations d'authentification. Cela permet de valider la configuration des accès à l'aide de l'API de gestion RabbitMQ. Pour plus d'informations, consultez la section Validation des accès.

Authentification et autorisation simples

Amazon MQ pour les utilisateurs du broker RabbitMQ

> **Note**
>
> Cette rubrique décrit la gestion des utilisateurs du broker avec le mécanisme d'authentification et d'autorisation interne par défaut de RabbitMQ. Pour plus d'informations sur toutes les méthodes d'authentification et d'autorisation prises en charge, consultez Amazon MQ pour l'authentification et l'autorisation RabbitMQ.

Chaque connexion client AMQP 0-9-1 est associée à un utilisateur. Cet utilisateur doit être authentifié. Chaque connexion client cible également un hôte virtuel (vhost). L'utilisateur doit disposer d'un ensemble d'autorisations pour cet hôte virtuel. Un utilisateur peut avoir l'autorisation de configurer, d'écrire dans, et de lire à partir des files d'attente et des échanges dans un vhost. Vous spécifiez les informations d'identification de l'utilisateur et le vhost cible lorsque la connexion est établie.

Lorsque vous créez un agent Amazon MQ pour RabbitMQ pour la première fois, Amazon MQ utilise les informations d'identification de connexion que vous fournissez pour créer un utilisateur RabbitMQ avec la balise `administrator`. Vous pouvez ensuite ajouter et gérer des utilisateurs via l'API de gestion RabbitMQ ou la console web RabbitMQ. Vous pouvez également utiliser la console web RabbitMQ ou l'API de gestion pour définir ou modifier les autorisations utilisateur et les balises.

> **Note**
>
> Les utilisateurs RabbitMQ ne seront pas stockés ou affichés via l'API Users Amazon MQ.

> **Important**
>
> Amazon MQ pour RabbitMQ ne prend pas en charge le nom d'utilisateur « invité » et supprimera le compte invité par défaut lorsque vous créerez un nouveau courtier. Amazon MQ supprimera également régulièrement tout compte créé par un client appelé « invité ».

Pour créer un nouvel utilisateur avec l'API de gestion RabbitMQ, utilisez le point de terminaison et le corps de requête de l'API suivants. Remplacez *username* et *password* par vos nouveaux identifiants de connexion.

```
PUT /api/users/username HTTP/1.1

    {"password":"password","tags":"administrator"}
```

Important

- N'ajoutez pas de données d'identification personnelle (PII) ou d'autres données confidentielles ou sensibles dans les noms d'utilisateur des agents. Les noms d'utilisateur des courtiers sont accessibles à d'autres AWS services, notamment aux CloudWatch journaux. Les noms d'utilisateur des agents ne sont pas destinées à être utilisées pour des données privées ou sensibles.
- Si vous perdez l'accès à tous les comptes d'administrateur, consultez la section Récupération de l'accès du courtier pour utiliser l'authentification IAM à des fins de restauration.

La clé `tags` est obligatoire et est une liste séparées par des virgules de balises pour l'utilisateur. Amazon MQ prend en charge les balises utilisateur `administrator`, `management`, `monitoring` et `policymaker`.

Vous pouvez définir des autorisations pour un utilisateur individuel à l'aide du point de terminaison et du corps de requête de l'API suivants. Remplacez *vhost* et *username* par vos informations. Pour le vhost par défaut `/`, utilisez `%2F`.

```
PUT /api/permissions/vhost/username HTTP/1.1

    {"configure":".*","write":".*","read":".*"}
```

Note

Les clés `configure`, `read` et `write` sont toutes obligatoires.

En utilisant la valeur `.*` du caractère générique, cette opération accorde à l'utilisateur des autorisations de lecture, d'écriture et de configuration pour toutes les files d'attente dans le vhost spécifié à l'utilisateur. Pour plus d'informations sur la gestion des utilisateurs via l'API de gestion RabbitMQ, consultez API HTTP de gestion RabbitMQ.

OAuth Authentification et autorisation 2.0 pour Amazon MQ pour RabbitMQ

Amazon MQ pour RabbitMQ prend en charge plusieurs méthodes d'authentification et d'autorisation. Pour plus d'informations sur toutes les méthodes prises en charge, consultez Authentification et autorisation pour Amazon MQ pour les courtiers RabbitMQ.

Dans le OAuth cadre de l'authentification et de l'autorisation 2.0, les utilisateurs du broker et leurs autorisations sont gérés par un fournisseur d'identité OAuth 2.0 externe (IdP). L'authentification des utilisateurs et les autorisations de ressources pour les hôtes virtuels, les échanges, les files d'attente et les sujets sont centralisées via le système de périmètre du fournisseur OAuth 2.0. Cela simplifie la gestion des utilisateurs et permet l'intégration aux systèmes d'identité existants.

⚠ Considérations importantes

- OAuth L'intégration 2.0 n'est pas prise en charge sur Amazon MQ pour les courtiers ActiveMQ.
- Amazon MQ pour RabbitMQ ne prend pas en charge les certificats de serveur émis par une autorité de certification privée.
- Le plugin RabbitMQ OAuth 2.0 ne prend pas en charge les points de terminaison d'introspection de jetons et les jetons d'accès opaques. Il n'effectue pas non plus de contrôles de révocation des jetons.
- Vous devez inclure l'autorisation IAM`mq:UpdateBrokerAccessConfiguration`, pour activer la OAuth version 2.0 sur les courtiers existants.
- Amazon MQ crée automatiquement un utilisateur du système nommé `monitoring-AWS-OWNED-DO-NOT-DELETE` avec des autorisations de surveillance uniquement. Cet utilisateur utilise le système d'authentification interne de RabbitMQ même sur les courtiers OAuth compatibles 2.0 et est limité à l'accès à l'interface de bouclage uniquement.

Pour plus d'informations sur la configuration de la OAuth version 2.0 pour votre Amazon MQ pour les courtiers RabbitMQ, consultez. Utilisation de l'authentification et de l'autorisation OAuth 2.0

Sur cette page

Configurations OAuth 2.0 prises en charge

Amazon MQ pour RabbitMQ prend en charge toutes les variables configurables dans le plug-in RabbitMQ OAuth 2.0, avec les exceptions suivantes :

- `auth_oauth2.https.cacertfile`
- `auth_oauth2.oauth_providers.{id/index}.https.cacertfile`
- `management.oauth_client_secret`

  Amazon MQ ne prenant pas en charge cette clé, nous ne prenons pas en charge l'UAA en tant qu'IdP.
- `management.oauth_resource_servers.{id/index}.oauth_client_secret`
- `auth_oauth2.signing_keys.{id/index}`

Validations supplémentaires pour l'authentification OAuth 2.0

Amazon MQ applique également les validations supplémentaires suivantes pour l'authentification 2.0 : OAuth

- Tout URLs doit commencer par`https://`.
- Algorithmes de signature pris en charge : `Ed25519` `Ed25519ph` `Ed448` `Ed448ph` `EdDSAES256K`,`ES256`,,`ES384`,`ES512`,`HS256`,`HS384`,`HS512`,`PS256`,`PS384`,`PS512`,,`RS256`,`RS384`, et`RS512`.

Authentification et autorisation IAM pour Amazon MQ pour RabbitMQ

Amazon MQ pour RabbitMQ prend en charge plusieurs méthodes d'authentification et d'autorisation. Pour plus d'informations sur toutes les méthodes prises en charge, consultez Authentification et autorisation pour Amazon MQ pour les courtiers RabbitMQ.

L'authentification et l'autorisation IAM permettent aux utilisateurs du broker de s'authentifier à l'aide des informations d'identification AWS IAM via la fédération sortante IAM. Dans cette méthode,

les informations d'identification IAM sont utilisées pour obtenir des jetons JWT auprès du AWS Security Token Service (STS). Ces jetons JWT servent de jetons OAuth 2.0 pour l'authentification, en tirant parti du support OAuth 2.0 existant dans Amazon MQ pour RabbitMQ, qui agit en tant que fournisseur d' AWS identité 2.0. OAuth AWS IAM gère l'authentification des utilisateurs, tandis que les autorisations de ressources pour les hôtes virtuels, les échanges, les files d'attente et les sujets sont gérées via des politiques IAM et des alias de portée configurés dans RabbitMQ.

⚠ Considérations importantes

- L'authentification IAM est prise en charge sur les versions 3.13, 4.2 et supérieures de RabbitMQ. Il n'est pas pris en charge sur Amazon MQ pour les courtiers ActiveMQ.
- L'authentification IAM nécessite que la fédération sortante IAM soit configurée et disponible dans votre compte. AWS
- Cette méthode s'appuie sur l'infrastructure OAuth 2.0 existante d'Amazon MQ pour RabbitMQ, en AWS servant de fournisseur d'identité 2.0. OAuth
- Amazon MQ crée automatiquement un utilisateur du système nommé `monitoring-AWS-OWNED-DO-NOT-DELETE` avec des autorisations de surveillance uniquement. Cet utilisateur utilise le système d'authentification interne de RabbitMQ, même sur les courtiers compatibles IAM, et est limité à l'accès à l'interface de bouclage uniquement.

Sur cette page

Comment fonctionne l'authentification IAM

L'authentification IAM pour Amazon MQ pour RabbitMQ utilise la fédération sortante IAM pour AWS permettre aux informations d'identification IAM de s'authentifier auprès des courtiers RabbitMQ. Les informations d'identification IAM sont utilisées pour obtenir des jetons JWT auprès du AWS Security Token Service (STS), et ces jetons JWT servent de jetons OAuth 2.0 pour l'authentification auprès du courtier RabbitMQ.

Limitations

L'authentification IAM pour Amazon MQ pour RabbitMQ présente les limites suivantes :

- Configuration de la réclamation de portée — Vous ne pouvez pas utiliser une revendication de portée directement car le jeton JWT de STS est imbriqué. L'essentiel est d'utiliser `sts.amazonaws.com` des alias de scope dans la configuration de RabbitMQ pour mapper les rôles IAM aux autorisations RabbitMQ. Cette limitation empêche également l'utilisation complète des politiques IAM pour l'autorisation, nécessitant plutôt la configuration de RabbitMQ pour l'autorisation.

Pour plus d'informations sur la configuration de l'authentification et de l'autorisation IAM pour vos courtiers Amazon MQ pour RabbitMQ, consultez. Utilisation de l'authentification et de l'autorisation IAM

Authentification et autorisation HTTP pour Amazon MQ pour RabbitMQ

Amazon MQ pour RabbitMQ prend en charge l'authentification et l'autorisation des utilisateurs du broker à l'aide d'un serveur HTTP externe. Pour connaître les autres méthodes prises en charge, consultez Authentification et autorisation pour Amazon MQ pour les courtiers RabbitMQ.

> **Note**
>
> Le plug-in d'authentification HTTP est uniquement disponible pour Amazon MQ pour RabbitMQ version 4 et supérieure.

> **Considérations importantes**
>
> - Le serveur HTTP doit être accessible via l'Internet public. Amazon MQ pour RabbitMQ peut être configuré pour s'authentifier auprès du serveur HTTP à l'aide du protocole TLS mutuel.
> - Amazon MQ pour RabbitMQ impose l'utilisation de AWS ARNs pour les paramètres qui nécessitent l'accès au système de fichiers local. Voir le support de l'ARN dans la configuration de RabbitMQ pour plus de détails.
> - Vous devez inclure l'autorisation IAM`mq:UpdateBrokerAccessConfiguration`, pour activer l'authentification HTTP sur les courtiers existants.
> - Amazon MQ crée automatiquement un utilisateur du système nommé `monitoring-AWS-OWNED-DO-NOT-DELETE` avec des autorisations de surveillance uniquement. Cet utilisateur utilise le système d'authentification interne de RabbitMQ même sur les courtiers

compatibles HTTP et est limité à l'accès à l'interface de bouclage uniquement. Amazon MQ empêche la suppression de cet utilisateur en ajoutant le tag utilisateur protégé.

Pour plus d'informations sur la configuration de l'authentification HTTP pour votre Amazon MQ pour les courtiers RabbitMQ, consultez. Utilisation de l'authentification et de l'autorisation HTTP

Sur cette page

Configurations HTTP prises en charge

Amazon MQ pour RabbitMQ prend en charge toutes les variables configurables dans le plug-in d'authentification HTTP RabbitMQ, avec les exceptions suivantes qui sont requises. AWS ARNs Pour plus de détails sur le support de l'ARN, voir le support de l'ARN dans la configuration de RabbitMQ.

Configurations nécessitant ARNs

`auth_http.ssl_options.cacertfile`

Utiliser `aws.arns.auth_http.ssl_options.cacertfile` à la place `auth_http.ssl_options.certfile`

Utiliser `aws.arns.auth_http.ssl_options.certfile` à la place `auth_http.ssl_options.keyfile`

Utiliser `aws.arns.auth_http.ssl_options.keyfile` à la place

Options SSL non prises en charge

Les options de configuration SSL suivantes ne sont pas non plus prises en charge :

Voir la liste complète

- `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`

Validations supplémentaires pour les configurations HTTP dans Amazon MQ

Amazon MQ applique également les validations supplémentaires suivantes pour l'authentification et l'autorisation HTTP :

- `auth_http.http_method`doit être `get` soit `post`
- Les configurations de chemin suivantes doivent utiliser le protocole HTTPS URLs :
  - `auth_http.user_path`
  - `auth_http.vhost_path`
  - `auth_http.resource_path`
  - `auth_http.topic_path`
- Si un paramètre nécessite l'utilisation d'un AWS ARN, `aws.arns.assume_role_arn` il doit être fourni.

Authentification par certificat SSL pour Amazon MQ pour RabbitMQ

Amazon MQ pour RabbitMQ prend en charge l'authentification des utilisateurs du broker à l'aide de certificats clients X.509. Pour connaître les autres méthodes prises en charge, consultez Authentification et autorisation pour Amazon MQ pour les courtiers RabbitMQ.

> **Note**
>
> Le plug-in d'authentification par certificat SSL est uniquement disponible pour Amazon MQ pour RabbitMQ version 4 et supérieure.

> **Considérations importantes**
>
> - Les certificats clients doivent être signés par une autorité de certification (CA) fiable. Amazon MQ pour RabbitMQ valide la chaîne de certificats lors de l'authentification.
> - Amazon MQ pour RabbitMQ impose l'utilisation de pour les paramètres liés aux certificats tels que les certificats CA et AWS ARNs pour les paramètres nécessitant un accès au système de fichiers local. Voir le support de l'ARN dans la configuration de RabbitMQ pour plus de détails.
> - Amazon MQ crée automatiquement un utilisateur du système nommé `monitoring-AWS-OWNED-DO-NOT-DELETE` avec des autorisations de surveillance uniquement. Cet utilisateur utilise le système d'authentification interne de RabbitMQ, même sur les courtiers dotés de certificats SSL, et est limité à l'accès à l'interface de bouclage uniquement. Amazon MQ empêche la suppression de cet utilisateur en ajoutant le tag utilisateur protégé.

Pour plus d'informations sur la configuration de l'authentification par certificat SSL pour votre Amazon MQ pour les courtiers RabbitMQ, consultez. Utilisation de l'authentification par certificat SSL

Sur cette page

Configurations SSL prises en charge

Amazon MQ pour RabbitMQ prend en charge SSL/TLS la configuration des connexions client. Pour plus de détails sur le support de l'ARN, voir le support de l'ARN dans la configuration de RabbitMQ.

Configurations nécessitant ARNs

`ssl_options.cacertfile`

Utiliser `aws.arns.ssl_options.cacertfile` à la place

Configurations de connexion par certificat SSL

Les configurations suivantes contrôlent la manière dont les noms d'utilisateur sont extraits des certificats clients :

`ssl_cert_login_from`

Spécifie le champ de certificat à utiliser pour l'extraction du nom d'utilisateur. Valeurs prises en charge :

- `distinguished_name`- Utilisez le nom distinctif complet
- `common_name`- Utilisez le champ Nom commun (CN)
- `subject_alternative_name`ou `subject_alt_name` - Utiliser le nom alternatif du sujet

`ssl_cert_login_san_type`

Lorsque vous utilisez le nom alternatif du sujet, spécifiez le type de SAN. Valeurs prises en charge : `dns`ip,`email`,`uri`, `other_name`

`ssl_cert_login_san_index`

Lorsque vous utilisez le nom alternatif du sujet, spécifie l'index de l'entrée SAN à utiliser (base zéro). Doit être un entier non négatif.

Options SSL pour les connexions client

Les options SSL suivantes s'appliquent aux connexions client :

`ssl_options.verify`

Mode de vérification par les pairs. Valeurs prises en charge :`verify_none`, `verify_peer`

`ssl_options.fail_if_no_peer_cert`

S'il faut rejeter les connexions si le client ne fournit pas de certificat. Valeur booléenne.

`ssl_options.depth`

Profondeur maximale de la chaîne de certificats pour la vérification.

`ssl_options.hostname_verification`

Mode de vérification du nom d'hôte. Valeurs prises en charge :`wildcard`, `none`

Options SSL non prises en charge

Les options de configuration SSL suivantes ne sont pas prises en charge :

Voir la liste complète

- `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`

Validations supplémentaires pour les configurations SSL dans Amazon MQ

Amazon MQ applique également les validations supplémentaires suivantes pour l'authentification par certificat SSL :

- Si un paramètre nécessite l'utilisation d'un AWS ARN, `aws.arns.assume_role_arn` il doit être fourni.

Authentification et autorisation LDAP pour Amazon MQ pour RabbitMQ

Amazon MQ pour RabbitMQ prend en charge l'authentification et l'autorisation des utilisateurs du broker à l'aide d'un serveur LDAP externe. Pour connaître les autres méthodes prises en charge, consultez Authentification et autorisation pour Amazon MQ pour les courtiers RabbitMQ.

⚠ Considérations importantes

- Le serveur LDAP doit être accessible via l'Internet public. Amazon MQ pour RabbitMQ peut être configuré pour s'authentifier auprès du serveur LDAP à l'aide du protocole TLS mutuel.
- Amazon MQ pour RabbitMQ impose l'utilisation de pour les paramètres LDAP sensibles tels que les mots de passe et AWS ARNs pour les paramètres nécessitant un accès au système de fichiers local. Voir le support de l'ARN dans la configuration de RabbitMQ pour plus de détails.
- Vous devez inclure l'autorisation IAM`mq:UpdateBrokerAccessConfiguration`, pour activer LDAP sur les courtiers existants.
- Amazon MQ crée automatiquement un utilisateur du système nommé `monitoring-AWS-OWNED-DO-NOT-DELETE` avec des autorisations de surveillance uniquement. Cet utilisateur utilise le système d'authentification interne de RabbitMQ même sur les courtiers compatibles LDAP et est limité à l'accès à l'interface de bouclage uniquement. Amazon MQ empêche la suppression de cet utilisateur en ajoutant le tag utilisateur protégé.

Pour plus d'informations sur la configuration du protocole LDAP pour votre Amazon MQ pour les courtiers RabbitMQ, consultez. Utilisation de l'authentification et de l'autorisation LDAP

Sur cette page

- Validations supplémentaires pour les configurations LDAP dans Amazon MQ

Configurations LDAP prises en charge

Amazon MQ pour RabbitMQ prend en charge toutes les variables configurables dans le plug-in LDAP RabbitMQ, avec les exceptions suivantes qui sont requises. AWS ARNs Pour plus de détails sur le support de l'ARN, voir le support de l'ARN dans la configuration de RabbitMQ.

Configurations nécessitant ARNs

`auth_ldap.dn_lookup_bind.password`

Utiliser `aws.arns.auth_ldap.dn_lookup_bind.password` à la place `auth_ldap.other_bind.password`

Utiliser `aws.arns.auth_ldap.other_bind.password` à la place `auth_ldap.ssl_options.cacertfile`

Utiliser `aws.arns.auth_ldap.ssl_options.cacertfile` à la place `auth_ldap.ssl_options.certfile`

Utiliser `aws.arns.auth_ldap.ssl_options.certfile` à la place `auth_ldap.ssl_options.keyfile`

Utiliser `aws.arns.auth_ldap.ssl_options.keyfile` à la place

Options SSL non prises en charge

Les options de configuration SSL suivantes ne sont pas non plus prises en charge :

Voir la liste complète

- `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`

Validations supplémentaires pour les configurations LDAP dans Amazon MQ

Amazon MQ applique également les validations supplémentaires suivantes pour l'authentification et l'autorisation LDAP :

- `auth_ldap.log`ne peut pas être défini sur `network_unsafe`
- Le serveur LDAP doit utiliser LDAPS. Activé `auth_ldap.use_ssl` ou `auth_ldap.use_starttls` doit être explicitement activé
- Si un paramètre nécessite l'utilisation d'un AWS ARN, `aws.arns.assume_role_arn` il doit être fourni.
- `auth_ldap.servers`doit être une adresse IP ou un FQDN valide
- Les clés suivantes doivent être un nom distinctif LDAP valide :
  - `auth_ldap.dn_lookup_base`
  - `auth_ldap.dn_lookup_bind.user_dn`
  - `auth_ldap.other_bind.user_dn`
  - `auth_ldap.group_lookup_base`

Plug-ins

Amazon MQ pour RabbitMQ prend également en charge les plug-ins suivants.

- Plugin de gestion RabbitMQ

- Plug-in pour pelle
- Plugin de fédération
- Plugin d'échange de hachage cohérent
- OAuth 2 plugins
- Plug-in LDAP
- Plug-in HTTP
- Plug-in de certificat SSL
- plugin aws
- Plug-in d'échange de sujets JMS

Plugin de gestion RabbitMQ

Amazon MQ pour RabbitMQ prend en charge le plug-in de gestion RabbitMQ, qui fournit une API de gestion basée sur HTTP ainsi qu'une interface utilisateur basée sur un navigateur pour la console Web RabbitMQ. Vous pouvez utiliser la console web et l'API de gestion pour créer et gérer des utilisateurs et des politiques de l'agent.

Plug-in Shovel

Amazon MQ pour RabbitMQ prend en charge le plugin RabbitMQ shovel, qui vous permet de déplacer des messages d'une file d'attente ou d'un échange d'un courtier à un autre. Vous pouvez utiliser la pelle pour connecter des agents à couplage faible et distribuer des messages loin des nœuds avec des charges de messages plus lourdes.

> **Important**
>
> Vous ne pouvez pas configurer de pelle entre les files d'attente ou les échanges si la destination de la pelle est un agent privé.
> Amazon MQ ne prend pas en charge l'utilisation de pelles statiques.

Seules les pelles dynamiques sont prises en charge. Les pelles dynamiques sont configurées à l'aide de paramètres d'exécution et peuvent être démarrées et arrêtées à tout moment par programmation via une connexion client. Par exemple, à l'aide de l'API de gestion RabbitMQ, vous pouvez créer une requête PUT vers le point de terminaison d'API suivant pour configurer une pelle dynamique.

Dans l'exemple, {vhost} peut être remplacé par le nom du vhost du broker, et {name} par le nom de la nouvelle pelle dynamique.

```
/api/parameters/shovel/{vhost}/{name}
```

Dans le corps de requête, vous devez spécifier soit une file d'attente, soit un échange, mais pas les deux. L'exemple ci-dessous configure une pelle dynamique entre une file locale spécifiée dans src-queue et une file distante définie dans dest-queue. De même, vous pouvez utiliser les paramètres src-exchange et dest-exchange pour configurer une pelle entre deux échanges.

```
{
"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 de fédération

Amazon MQ prend en charge les échanges fédérés et les files d'attente à l'aide du plug-in de fédération RabbitMQ. Avec la fédération, vous pouvez répliquer le flux de messages entre les files d'attente, les échanges et les consommateurs sur des agents distincts. Les files d'attente fédérées et les échanges utilisent point-to-point des liens pour entrer en contact avec les pairs d'autres courtiers. Alors que les échanges fédérés, par défaut, acheminent les messages une fois, les files d'attente fédérées peuvent déplacer des messages n'importe quel nombre de fois selon les besoins des consommateurs.

Vous pouvez utiliser la fédération pour autoriser un agent en aval pour consommer un message provenant d'un échange ou d'une file d'attente en amont. Vous pouvez activer la fédération sur les agents en aval à l'aide de la console web RabbitMQ ou de l'API de gestion.

> **Important**
>
> Vous ne pouvez pas configurer de fédération si la file d'attente ou l'échange en amont se trouve dans un agent privé. Vous pouvez uniquement configurer de fédération entre les files

d'attente ou les échanges dans des agents publics, ou entre une file d'attente ou un échange en amont dans un agent public, et une file d'attente ou un échange en aval dans un agent privé.

Par exemple, l'API de gestion vous permet de configurer la fédération en procédant comme suit.

- Configurez un ou plusieurs flux en amont qui définissent les connexions de fédération à d'autres nœuds. Vous pouvez définir des connexions de fédération à l'aide de la console web RabbitMQ ou de l'API de gestion. À l'aide de l'API de gestion, vous pouvez créer une requête POST vers/api/parameters/federation-upstream/%2f/myupstreamavec le corps de requête suivant.

```
{"value":{"uri":"amqp://server-name","expires":3600000}}
```

- Configurez une politique pour permettre à vos files d'attente ou échanges d'être fédérés. Vous pouvez configurer des politiques à l'aide de la console web RabbitMQ ou de l'API de gestion. À l'aide de l'API de gestion, vous pouvez créer une requête POST pour/api/policies/%2f/federate-me avec le corps de requête suivant.

```
{"pattern":"^amq\.", "definition":{"federation-upstream-set":"all"}, "apply-
to":"exchanges"}
```

> **Note**
>
> Le corps de la requête suppose que les échanges sur le serveur sont nommés en commençant par amq. L'utilisation de l'expression régulière ^amq \. garantira que la fédération est activée pour tous les échanges dont le nom commence par « amq ». Les échanges sur votre serveur RabbitMQ peuvent être nommés différemment.

Plugin d'échange de hachage cohérent

Amazon MQ pour RabbitMQ prend en charge le plug-in RabbitMQ Consistent Hash Exchange Type. Les échanges de hachage cohérent acheminent les messages vers les files d'attente en fonction d'une valeur de hachage calculée à partir de la clé de routage d'un message. Compte tenu d'une clé de routage raisonnablement uniforme, les échanges de hachage cohérents peuvent répartir les messages entre les files d'attente de manière raisonnablement uniforme.

Pour les files d'attente liées à un échange de hachage cohérent, la clé de liaison est a number-as-a-string qui détermine le poids de liaison de chaque file d'attente. Les files d'attente avec un poids de liaison plus élevé recevront une distribution proportionnellement plus élevée de messages provenant de l'échange de hachage cohérent auquel elles sont liées. Dans une topologie d'échange de hachage cohérent, les éditeurs peuvent simplement publier des messages dans l'échange, mais les consommateurs doivent être explicitement configurés pour consommer des messages provenant de files d'attente spécifiques.

OAuth Plug-in 2.0

Amazon MQ pour RabbitMQ prend en charge le plug-in principal d'authentification à 2 niveaux. OAuth Ce plugin est activé de manière conditionnelle en fonction de la configuration de votre courtier. Lorsqu'il est activé, ce plugin fournit une authentification et une autorisation OAuth 2.0 avec intégration à des fournisseurs d'identité OAuth 2.0 externes pour une gestion centralisée des utilisateurs et un contrôle d'accès. Pour plus d'informations sur l'authentification OAuth 2.0, consultezOAuth Authentification et autorisation 2.0.

Plug-in LDAP

Amazon MQ pour RabbitMQ prend en charge le plug-in principal d'authentification LDAP. Ce plugin est activé de manière conditionnelle en fonction de la configuration de votre courtier. Lorsqu'il est activé, ce plugin fournit une authentification et une autorisation LDAP avec intégration à des services d'annuaire LDAP externes pour une authentification et une autorisation centralisées des utilisateurs. Pour plus d'informations sur l'authentification LDAP, consultezAuthentification et autorisation LDAP.

Plug-in HTTP

Amazon MQ pour RabbitMQ prend en charge le plugin principal d'authentification HTTP. Ce plugin est activé de manière conditionnelle en fonction de la configuration de votre courtier. Lorsqu'il est activé, ce plugin fournit une authentification et une autorisation HTTP avec intégration à des serveurs HTTP externes pour une authentification et une autorisation centralisées des utilisateurs. Pour plus d'informations sur l'authentification HTTP, consultezAuthentification et autorisation HTTP.

> **Note**
>
> Le plug-in d'authentification HTTP est uniquement disponible pour Amazon MQ pour RabbitMQ version 4 et supérieure.

Plug-in de certificat SSL

Amazon MQ prend en charge le protocole TLS mutuel (MTL) pour les courtiers RabbitMQ. Le plugin d'authentification SSL utilise des certificats clients issus de connexions mTLS pour authentifier les utilisateurs. Ce plugin est activé de manière conditionnelle en fonction de la configuration de votre courtier. Lorsqu'il est activé, il fournit une authentification basée sur des certificats à l'aide de certificats clients X.509 pour une authentification renforcée sans transmettre d'informations d'identification sur le réseau. Pour plus d'informations sur l'authentification par certificat SSL, consultezAuthentification par certificat SSL.

> **Note**
>
> Le plug-in d'authentification par certificat SSL est uniquement disponible pour Amazon MQ pour RabbitMQ version 4 et supérieure.

plugin aws

Le plugin aws est activé de manière conditionnelle par Amazon MQ pour RabbitMQ en fonction de la configuration de votre courtier. Ce plugin communautaire, développé et maintenu par Amazon MQ, permet de récupérer en toute sécurité les informations d'identification et les certificats des AWS services utilisés AWS ARNs dans les paramètres de configuration de RabbitMQ. Pour plus d'informations sur la prise en charge de l'ARN, consultezARN support in RabbitMQ configuration.

Plug-in d'échange de sujets JMS

Le plugin JMS Topic Exchange est toujours activé par Amazon MQ pour RabbitMQ. Il fonctionne avec le client JMS RabbitMQ pour permettre aux applications JMS nouvelles et existantes de se connecter à Amazon MQ pour RabbitMQ.

> **Note**
>
> Le plugin JMS Topic Exchange est uniquement disponible pour Amazon MQ pour RabbitMQ version 4 et supérieure. Il est activé par défaut mais ne s'active que lorsque le client RabbitMQ JMS est utilisé pour exécuter des charges de travail JMS.

Protocoles pris en charge

Vous pouvez accéder à vos courtiers RabbitMQ en utilisant n'importe quel langage de programmation pris en charge par RabbitMQ et en activant le protocole TLS pour l'une des spécifications de protocole suivantes :

- AMQP (0-9-1)
- AMQP 1.0
- JMS 1.1
- JMS 2.0
- JMS 3.1

Support Amazon MQ pour RabbitMQ JMS

Vous pouvez désormais exécuter des charges de travail JMS 1.1, 2.0 et 3.1 sur Amazon MQ pour RabbitMQ 4 avec le client JMS RabbitMQ.

Client JMS RabbitMQ

Le client RabbitMQ JMS est une bibliothèque client JMS open source dont vous avez besoin pour connecter votre application JMS aux courtiers Amazon MQ RabbitMQ. Pour plus d'informations, veuillez consulter le GitHub référentiel officiel.

JMS 1.1, 2.0 et 3.1 pris en charge APIs

À partir d'Amazon MQ pour RabbitMQ 4, le plugin est toujours activé. `jms-topic-exchange` Par conséquent, vous pouvez utiliser Amazon MQ pour RabbitMQ 4 et le client RabbitMQ JMS pour votre charge de travail JMS. Tous les JMS APIs définis dans le JMS 1.1 sont pris en charge, sauf :

- APIs Les sessions de serveur ne sont pas prises en charge.
- APIs Les transactions XA ne sont pas prises en charge.
- Le sélecteur JMS pour la destination de la file d'attente JMS n'est pas pris en charge.
- L'attribut `NoLocal` d'abonnement JMS n'est pas pris en charge.

Tous les nouveaux ajouts APIs dans JMS 2.0 et JMS 3.1 sont pris en charge, à l'exception des suivants :

- `JMSProducer.setDeliveryDelay`L'API n'est pas prise en charge.

Pour en savoir plus sur la connexion de votre application JMS au courtier Amazon MQ pour RabbitMQ, consultez le didacticiel sur la connexion de votre application JMS au courtier Amazon MQ pour RabbitMQ

Authentification et autorisation

Tous les mécanismes d'authentification et d'autorisation répertoriés dans cette section sont pris en charge. Les informations d'identification utilisées pour vous connecter au courtier à l'aide du client JMS sont les mêmes que si vous vous connectiez au courtier RabbitMQ à l'aide d'un client Java AMQP.

Interopérabilité avec les files d'attente AMQP sur RabbitMQ

Vous pouvez utiliser le client JMS RabbitMQ pour envoyer des messages JMS à un échange AMQP et consommer les messages d'une file d'attente AMQP (cette fonctionnalité ne prend pas en charge les rubriques JMS). Cela vous permet d'interopérer ou de migrer certaines charges de travail JMS vers des charges de travail AMQP. Pour plus d'informations, veuillez consulter la documentation officielle du client.

Appliquer des politiques à Amazon MQ pour RabbitMQ

Vous pouvez appliquer des politiques et des limites personnalisées avec les valeurs par défaut recommandées par Amazon MQ. Si vous avez supprimé les politiques et limites par défaut recommandées et que vous souhaitez les recréer, ou si vous avez créé des vhosts supplémentaires et souhaitez appliquer les politiques et limites par défaut à vos nouveaux vhosts, vous pouvez suivre les étapes suivantes.

> **Important**
>
> Sur Amazon MQ pour les versions 3.13 et antérieures du moteur RabbitMQ, la politique d'opérateur par défaut actuelle est la suivante :

```
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
```

Sur les versions 4.0 et supérieures, la politique d'opérateur par défaut est passée à :

```
vhost name pattern apply-to definition priority/
 default_operator_policy_AWS_managed .* classic_queues {"queue-version":2} 0
```

Cette modification est nécessaire car la mise en miroir de files d'attente classique et les paramètres de politique HA ne sont pas pris en charge dans RabbitMQ 4.
Vous ne pouvez pas créer de politique qui s'applique à la fois aux files d'attente classiques en miroir et aux files d'attente de quorum. Si vous souhaitez que votre politique ne s'applique qu'aux files d'attente du quorum, vous devez la `--apply-to` définir `quorum_queues` sur.
Si vous utilisez des files d'attente en miroir classiques et des files d'attente de quorum, vous devez créer une politique distincte `--apply-to:classic_queues` ainsi qu'une politique de files d'attente de quorum.

Important

Pour effectuer les opérations suivantes, vous devez disposer d'un utilisateur d'agent Amazon MQ for RabbitMQ avec des autorisations d'administrateur. Vous pouvez utiliser l'utilisateur administrateur créé lors de la création de l'agent pour la première fois, ou un autre utilisateur que vous avez créé par la suite. Le tableau suivant fournit la balise utilisateur administrateur et les autorisations nécessaires en tant que modèles d'expression régulière (regexp).

Étiquettes	Lire regexp	Configurer regexp	Regexp en écriture
`administrator`	`.*`	`.*`	`.*`

Pour plus d'informations sur la création d'utilisateurs RabbitMQ et sur la gestion des balises utilisateur et des autorisations, consultez Amazon MQ pour les utilisateurs du broker RabbitMQ.

Pour appliquer des politiques par défaut et des limites d'hôte virtuel à l'aide de la console web RabbitMQ

1. Connectez-vous à la console Amazon MQ.
2. Dans le panneau de navigation de gauche, choisissez Brokers (Agents).
3. Dans la liste des agents, choisissez le nom de l'agent auquel vous souhaitez appliquer la nouvelle politique.
4. Sur la page de détails de l'agent, dans la section Connections (Connexions), choisissez l'URL de console web RabbitMQ. La console web RabbitMQ s'ouvre dans un nouvel onglet ou une nouvelle fenêtre du navigateur.
5. Connectez-vous à la console web RabbitMQ à l'aide du nom d'utilisateur et du mot de passe de l'administrateur de l'agent.
6. Dans la console web RabbitMQ, en haut de la page, choisissez Admin (Administrateur).
7. Dans la page Admin (Administrateur), dans le volet de navigation droit, choisissez Policies (Politiques).
8. Dans la page Policies (Politiques), vous pouvez consulter la liste actuelle User policies (Politiques utilisateur) de l'agent. Sous User policies (Politiques utilisateur), développez Add/update a policy (Ajouter ou mettre à jour une politique).
9. Pour créer une politique d'agent, sous Add/update a policy (Ajouter ou mettre à jour une politique), procédez comme suit :

    a. Pour Virtual Host (Hôte virtuel), choisissez le nom du vhost auquel vous souhaitez attacher les politiques à partir de la liste déroulante. Pour choisir le vhost par défaut, choisissez /.

    > **Note**
    >
    > Si vous n'avez pas créé de vhosts supplémentaires, l'option Virtual Host (Hôte virtuel) n'est pas affichée dans la console RabbitMQ et les politiques sont appliquées uniquement au vhost par défaut.

    b. Dans Name (Name), attribuez un nom à votre politique IAM, par exemple **`policy-defaults`**.

    c. Pour Pattern (Modèle), entrez le modèle regexp **`.*`** afin que la politique corresponde à toutes les files d'attente sur l'agent.

    d. Pour Apply to (Appliquer à), choisissez Exchanges and queues (Échanges et files d'attente) dans la liste déroulante.

e. Pour Priority (Priorité), saisissez un entier supérieur à toutes les autres politiques appliquées au vhost. Vous pouvez appliquer exactement un ensemble de définitions de politiques aux files d'attente et aux échanges RabbitMQ à tout moment. RabbitMQ choisit la politique correspondante avec la valeur de priorité la plus élevée. Pour plus d'informations sur les priorités de politique et sur la façon de combiner des politiques, veuillez consulter Policies (Politiques) dans la documentation du serveur RabbitMQ.

f. Pour Definition (Définition), ajoutez les paires clé-valeur suivantes :

- **`queue-mode=lazy`**. Choisissez String (Chaîne) dans la liste déroulante.
- **`overflow=reject-publish`**. Choisissez String (Chaîne) dans la liste déroulante.

> **Note**
>
> Ne s'applique pas aux agents à instance unique.

- **`max-length=`*`number-of-messages`***. *`number-of-messages`*Remplacez-la par la valeur recommandée par Amazon MQ en fonction de la taille de l'instance et du mode de déploiement du courtier, par exemple **`8000000`** pour un `mq.m7g.large` cluster. Choisissez Number (Numéro) dans la liste déroulante.

> **Note**
>
> Ne s'applique pas aux agents à instance unique.

g. Choisissez Add / update policy (Ajouter/mettre à jour une politique).

10. Confirmez que la nouvelle politique apparaît dans la liste User policies (Politiques utilisateur).

> **Note**
>
> Pour les agents en cluster, Amazon MQ applique automatiquement les définitions de politique `ha-mode: all` et `ha-sync-mode: automatic`.

11. Dans le panneau de navigation droit, cliquez sur Limits (Limites).
12. Dans la page Limits (limites), vous pouvez consulter une liste des limites d'hôte virtuel actuelles de l'agent. En dessous de Virtual host limits (Limites d'hôte virtuel), développez Set / update a virtual host limit (Définir/mettre à jour une limite d'hôte virtuel).

13. Pour créer une nouvelle limite de vhost, sous Set / update a virtual host limit (Définir/mettre à jour une limite d'hôte virtuel), procédez comme suit :

    a. Pour Virtual Host (Hôte virtuel) choisissez dans la liste déroulante le nom du vhost auquel vous souhaitez attacher les politiques. Pour choisir le vhost par défaut, choisissez /.

    b. Pour Limit (Limite), choisissez max-connections dans les options de la liste déroulante.

    c. Pour Value (Valeur), entrez la valeur Amazon MQ recommandée en fonction de la taille de l'instance et du mode de déploiement de l'agent, par exemple, **`15000`** pour un cluster `mq.m5.large`.

    d. Choisissez Set / update limit (Définir/mettre à jour la limite).

    e. Répétez les étapes ci-dessus et pour Limit (Limite), choisissez max-queues dans les options de la liste déroulante.

14. Confirmez que les nouvelles limites apparaissent dans la liste des limites d'hôte virtuel.

Pour appliquer des politiques et des limites d'hôte virtuel par défaut à l'aide de l'API de gestion RabbitMQ

1. Connectez-vous à la console Amazon MQ.
2. Dans le panneau de navigation de gauche, choisissez Brokers (Agents).
3. Dans la liste des agents, choisissez le nom de l'agent auquel vous souhaitez appliquer la nouvelle politique.
4. Sur la page de l'agent, dans la section Connection (Connexion), notez l'URL de la console web RabbitMQ. Il s'agit du point de terminaison de l'agent que vous utilisez dans une requête HTTP.
5. Ouvrez une nouvelle fenêtre de terminal ou de ligne de commande.
6. Pour créer une politique d'agent, entrez la commande `curl` suivante. Cette commande suppose la présence d'une file d'attente sur le vhost `/` par défaut, qui est encodé en tant que `%2F`. Pour appliquer la politique à un autre vhost, remplacez `%2F` par le nom du vhost.

> **Note**
>
> Remplacez *`username`* et *`password`* par vos identifiants de connexion d'administrateur. *`number-of-messages`*Remplacez-la par la valeur recommandée par Amazon MQ en fonction de la taille de l'instance et du mode de déploiement du courtier. *`policy-`*

*name*Remplacez-le par le nom de votre police. *broker-endpoint*Remplacez-le par l'URL que vous avez indiquée précédemment.

```
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. Pour confirmer que la nouvelle politique est ajoutée aux politiques utilisateur de votre agent, entrez la commande `curl` suivante pour répertorier toutes les politiques de l'agent.

```
curl -i -u username:password broker-endpoint/api/policies
```

8. Pour créer de nouvelles limites d'hôte virtuel `max-connections`, entrez les commandes `curl`. Cette commande suppose la présence d'une file d'attente sur le vhost `/` par défaut, qui est encodé en tant que `%2F`. Pour appliquer la politique à un autre vhost, remplacez `%2F` par le nom du vhost.

> **Note**
>
> Remplacez *username* et *password* par vos identifiants de connexion d'administrateur. *max-connections*Remplacez-la par la valeur recommandée par Amazon MQ en fonction de la taille de l'instance et du mode de déploiement du courtier. Remplacez le point de terminaison de l'agent par l'URL que vous avez notée précédemment.

```
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. Pour créer une nouvelle limite d'hôte virtuel `max-queues`, répétez l'étape précédente, mais modifiez la commande curl comme illustré dans ce qui suit.

```
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. Pour confirmer que les nouvelles limites sont ajoutées aux limites d'hôtes virtuels de votre agent, entrez la commande `curl` suivante pour répertorier toutes les limites des hôtes virtuels de l'agent.

```
curl -i -u username:password broker-endpoint/api/vhost-limits
```

Files d'attente de quorum pour RabbitMQ sur Amazon MQ

Les files d'attente de quorum sont un type de file d'attente répliquée composé d'un leader (réplique principale) et d'abonnés (autres répliques). Si le leader devient indisponible, les files d'attente de quorum utilisent l'algorithme de consensus Raft pour élire un nouveau nœud leader à la majorité des voix, et le leader précédent est rétrogradé au rang de nœud suiveur dans le même cluster. Les abonnés restants continuent de se répliquer comme avant. Chaque nœud se trouvant dans une zone de disponibilité différente, si un nœud est temporairement indisponible, la livraison des messages se poursuit avec la réplique du leader nouvellement élu dans une autre zone de disponibilité.

Les files d'attente de quorum sont utiles pour traiter les messages toxiques, qui apparaissent lorsqu'un message échoue et est mis en attente plusieurs fois.

Vous ne devez pas utiliser les files d'attente du quorum si vous :

- utiliser des files d'attente transitoires
- vous avez de longs arriérés de files d'attente
- privilégiez la faible latence

Pour déclarer une file d'attente de quorum, définissez l'en-tête `x-queue-type` sur`quorum`.

Rubriques

- Migration des files d'attente classiques vers des files d'attente de quorum sur Amazon MQ pour RabbitMQ
- Configurations des politiques pour les files d'attente de quorum pour Amazon MQ pour RabbitMQ
- Bonnes pratiques pour les files d'attente de quorum pour Amazon MQ pour RabbitMQ

Migration des files d'attente classiques vers des files d'attente de quorum sur Amazon MQ pour RabbitMQ

Vous pouvez migrer vos files d'attente miroir classiques vers des files d'attente de quorum sur les courtiers Amazon MQ sur la version 3.13 ou supérieure en créant un nouvel hôte virtuel sur le même cluster ou en effectuant une migration sur place.

Option 1 : migration des files d'attente en miroir classiques vers des files d'attente de quorum avec un nouvel hôte virtuel

Vous pouvez migrer vos files d'attente miroir classiques vers des files d'attente de quorum sur les courtiers Amazon MQ sur la version 3.13 ou supérieure en créant un nouvel hôte virtuel sur le même cluster.

1. Dans votre cluster existant, créez un nouvel hôte virtuel (vhost) avec le type de file d'attente par défaut comme quorum.
2. Créez le Plugin de fédération depuis le nouvel hôte virtuel avec l'URI pointant vers l'ancien hôte virtuel à l'aide de files d'attente miroir classiques.
3. En utilisant`rabbitmqadmin`, exportez les définitions de l'ancien hôte virtuel vers un nouveau fichier. Vous devez apporter des modifications au fichier de schéma afin qu'il soit compatible avec les files d'attente du quorum. Pour obtenir la liste complète des modifications que vous devez apporter au fichier, consultez la section Déplacement des définitions dans la documentation sur les files d'attente de quorum de RabbitMQ. Après avoir apporté les modifications nécessaires au fichier, réimportez les définitions sur le nouvel hôte virtuel.
4. Créez une nouvelle politique dans le nouveau vhost. Pour obtenir des recommandations sur les configurations des politiques Amazon MQ pour les files d'attente de quorum, consultez. Configurations des politiques pour les files d'attente de quorum pour Amazon MQ pour RabbitMQ Ensuite, lancez la fédération que vous avez créée précédemment depuis l'ancien hôte virtuel vers le nouveau serveur virtuel.
5. Dirigez les consommateurs et les producteurs vers le nouvel hôte virtuel.
6. Configurez le plug-in Shovel pour déplacer tous les messages restants. Une fois qu'une file d'attente est vide, supprimez la pelle.

Migration des files d'attente en miroir classiques vers des files d'attente de quorum en place

Vous pouvez migrer vos files d'attente miroir classiques vers des files d'attente de quorum sur Amazon MQ Brokers sur la version 3.13 ou supérieure en effectuant la migration sur place.

1. Arrêtez les consommateurs et les producteurs.
2. Créez une nouvelle file d'attente temporaire pour le quorum.
3. Configurez le plug-in Shovel pour déplacer tous les messages de l'ancienne file d'attente miroir classique vers la nouvelle file d'attente de quorum temporaire. Une fois que tous les messages ont été déplacés vers la file d'attente temporaire du quorum, supprimez le Shovel.
4. Supprimez la file d'attente miroir classique source. Recréez ensuite une file d'attente de quorum portant le même nom et les mêmes liaisons que la file d'attente miroir classique source.
5. Créez un nouveau Shovel pour déplacer les messages de la file d'attente de quorum temporaire vers la nouvelle file d'attente de quorum.

Configurations des politiques pour les files d'attente de quorum pour Amazon MQ pour RabbitMQ

Vous pouvez ajouter des configurations de politiques spécifiques aux files d'attente de quorum pour votre courtier RabbitMQ sur Amazon MQ.

Lorsque vous créez une politique pour les files d'attente de quorum, vous devez effectuer les opérations suivantes :

- Supprimez tous les attributs de stratégie commençant par`ha-mode`, `ha-paramsha-sync-mode`,`ha-sync-batch-size`,`ha-promote-on-shutdown`, et`ha-promote-on-failure`. `ha`
- Supprimez `queue-mode`.
- Modifier le trop-plein lorsqu'il est réglé sur `reject-publish-dlx`

> **Important**
>
> Amazon MQ pour RabbitMQ applique la totalité ou aucun des attributs d'une politique. Vous ne pouvez pas créer de politique qui s'applique à la fois aux files d'attente classiques en miroir et aux files d'attente de quorum. Si vous souhaitez que votre politique ne s'applique qu'aux files d'attente du quorum, vous devez la `--apply-to` définir `quorum_queues` sur.

Si vous utilisez des files d'attente en miroir classiques et des files d'attente de quorum, vous devez créer une politique distincte avec `--apply-to` : `classic_queues` ainsi qu'une politique de files d'attente de quorum.

Il n'est pas nécessaire de modifier `AWS-DEFAULT` les politiques car elles adoptent automatiquement le nouveau type de file d'attente dans le paramètre « s'applique à ». Pour plus d'informations sur les politiques par défaut d'Amazon MQ pour RabbitMQ, consultez. Configuration des politiques de l'opérateur

Bonnes pratiques pour les files d'attente de quorum pour Amazon MQ pour RabbitMQ

Nous vous recommandons d'utiliser les meilleures pratiques suivantes pour améliorer les performances lorsque vous travaillez avec des files d'attente de quorum.

Gérer les messages empoisonnés en fixant une limite de diffusion

Les messages empoisonnés apparaissent lorsqu'un message échoue et est renvoyé plusieurs fois. Vous pouvez définir une limite de remise de messages à l'aide de l'argument `delivery-limit` policy pour supprimer les messages redistribués plusieurs fois. Si un message est redistribué plus de fois que la limite de livraison ne le permet, le message est ensuite déposé et supprimé par RabbitMQ. Lorsque vous définissez une limite de livraison, le message est placé en file d'attente en tête de file d'attente.

Priorité des messages pour les files d'attente du quorum

Les files d'attente du quorum ne sont pas prioritaires pour les messages. Si vous avez besoin de la priorité des messages, vous devez créer plusieurs files d'attente de quorum. Pour plus d'informations sur la hiérarchisation des messages comportant plusieurs files d'attente de quorum, consultez la section Priorité des messages dans la documentation de RabbitMQ.

Utilisation du facteur de réplication par défaut

Amazon MQ pour RabbitMQ utilise par défaut un facteur de réplication de trois (3) nœuds pour les courtiers de clusters utilisant des files d'attente de quorum. Si vous apportez des modifications à`x-quorum-initial-group-size`, Amazon MQ utilisera à nouveau par défaut le facteur de réplication de 3.

Bonnes pratiques Amazon MQ for RabbitMQ

Suivez ces directives de préparation à la production pour optimiser les performances des courtiers et optimiser l'efficacité du débit des messages lorsque vous travaillez avec Amazon MQ pour les courtiers RabbitMQ.

> **Important**
>
> Actuellement, Amazon MQ ne prend pas en charge des flux, ni l'utilisation de la journalisation structurée dans JSON, introduite dans RabbitMQ 3.9.x.

Rubriques

- Meilleures pratiques pour la configuration des courtiers et la gestion des connexions dans Amazon MQ pour RabbitMQ
- Meilleures pratiques en matière de durabilité et de fiabilité des messages dans Amazon MQ pour RabbitMQ
- Meilleures pratiques en matière d'optimisation des performances et d'efficacité dans Amazon MQ pour RabbitMQ
- Meilleures pratiques en matière de résilience et de surveillance du réseau dans Amazon MQ pour RabbitMQ

Meilleures pratiques pour la configuration des courtiers et la gestion des connexions dans Amazon MQ pour RabbitMQ

La configuration du broker et la gestion des connexions constituent la première étape pour éviter les problèmes liés au débit des messages du broker, à l'utilisation des ressources et à la capacité à gérer les charges de travail de production. Lors de la création et de la configuration d'un courtier Amazon MQ pour RabbitMQ, suivez les bonnes pratiques suivantes pour sélectionner les types d'instances appropriés, gérer efficacement les connexions et configurer le préchargement des messages afin d'optimiser les performances de votre courtier.

> **Important**
>
> Amazon MQ pour RabbitMQ ne prend pas en charge le nom d'utilisateur « invité » et supprimera le compte invité par défaut lorsque vous créerez un nouveau courtier. Amazon MQ supprimera également régulièrement tout compte créé par un client appelé « invité ».

Étape 1 : Utiliser les déploiements de clusters

Pour les charges de travail de production, nous recommandons d'utiliser des déploiements en cluster plutôt que des courtiers à instance unique afin de garantir la haute disponibilité et la résilience des messages. Les déploiements en cluster suppriment les points de défaillance uniques et offrent une meilleure tolérance aux pannes.

Les déploiements de clusters se composent de trois nœuds de courtage RabbitMQ répartis sur trois zones de disponibilité, ce qui permet un basculement automatique et garantit la continuité des opérations même si une zone de disponibilité complète devient indisponible. Amazon MQ réplique automatiquement les messages sur tous les nœuds pour garantir la disponibilité en cas de panne ou de maintenance des nœuds.

Les déploiements de clusters sont essentiels pour les environnements de production et sont pris en charge par le contrat de niveau de service Amazon MQ.

Pour plus d'informations, consultez Déploiement de clusters dans Amazon MQ pour RabbitMQ.

Étape 2 : Choisissez le type d'instance de courtier approprié

Le débit de messages d'un type d'instance de courtier dépend du cas d'utilisation de votre application. `M7g.mediumne` doit être utilisé que pour tester les performances de l'application. L'utilisation de cette petite instance avant d'utiliser des instances plus grandes en production peut améliorer les performances des applications. Sur les types d'instance `m7g.large` et supérieurs, vous pouvez utiliser des déploiements de clusters pour garantir une haute disponibilité et une durabilité des messages. Les types d'instances de broker de plus grande taille peuvent gérer les niveaux de production des clients et des files d'attente, le haut débit, les messages en mémoire et les messages redondants.

Pour plus d'informations sur le choix du type d'instance approprié, consultez les directives de dimensionnement dans Amazon MQ pour RabbitMQ.

Étape 3 : Utiliser les files d'attente du quorum

Les files d'attente de quorum, associées au déploiement en cluster, devraient être le choix par défaut pour les types de files d'attente répliqués dans les environnements de production pour les courtiers RabbitMQ sur 3.13 et versions ultérieures. Les files d'attente de quorum sont un type de file d'attente répliqué moderne qui offre une fiabilité élevée, un débit élevé et une latence stable.

Les files d'attente de quorum utilisent l'algorithme de consensus Raft pour améliorer la tolérance aux pannes. Lorsque le nœud leader devient indisponible, les files d'attente du quorum élisent automatiquement un nouveau leader par un vote majoritaire, garantissant ainsi la transmission des messages avec un minimum de perturbations. Comme chaque nœud se trouve dans une zone de disponibilité différente, votre système de messagerie reste disponible même si une zone de disponibilité complète devient temporairement indisponible.

Pour déclarer une file d'attente de quorum, définissez l'en-tête sur `quorum` lors `x-queue-type` de la création de vos files d'attente.

Pour plus d'informations sur les files d'attente de quorum, y compris les stratégies de migration et les meilleures pratiques, consultez la section Queues de quorum dans Amazon MQ pour RabbitMQ.

Étape 4 : Utiliser plusieurs canaux

Pour éviter toute perte de connexion, utilisez plusieurs canaux sur une seule connexion. Les applications doivent éviter un rapport connexion/canal 1:1. Nous recommandons d'utiliser une connexion pour chaque processus, puis un canal pour chaque thread. Évitez l'utilisation excessive des canaux pour éviter les fuites.

Meilleures pratiques en matière de durabilité et de fiabilité des messages dans Amazon MQ pour RabbitMQ

Avant de passer votre application en production, suivez les bonnes pratiques suivantes pour éviter la perte de messages et la surutilisation des ressources.

Étape 1 : Utiliser des messages persistants et des files d'attente durables

Les messages persistants peuvent contribuer à protéger la durabilité des données en cas de panne ou de redémarrage d'un broker. Les messages persistants sont écrits sur le disque dès leur arrivée. Cependant, contrairement aux files d'attente paresseuses, les messages persistants sont mis en

cache à la fois dans la mémoire et dans le disque, sauf si l'agent a besoin de plus de mémoire. Dans les cas où plus de mémoire est nécessaire, les messages sont supprimés de la mémoire par le mécanisme d'agent RabbitMQ qui gère le stockage des messages sur disque, communément appelé couche de persistance.

Pour activer la persistance des messages, vous pouvez déclarer vos files d'attente comme `durable` et définissez le mode de remise des messages sur `persistent`. L'exemple suivant illustre l'utilisation de la bibliothèque client Java RabbitMQ pour déclarer une file d'attente durable. Lorsque vous travaillez avec AMQP 0-9-1, vous pouvez marquer les messages comme persistants en définissant le mode de livraison « 2 ».

```
boolean durable = true;
channel.queueDeclare("my_queue", durable, false, false, null);
```

Une fois que vous avez configuré votre file d'attente comme durable, vous pouvez envoyer un message persistant à votre file d'attente en définissant `MessageProperties` sur `PERSISTENT_TEXT_PLAIN` illustré dans l'exemple suivant.

```
import com.rabbitmq.client.MessageProperties;

channel.basicPublish("", "my_queue",
            MessageProperties.PERSISTENT_TEXT_PLAIN,
            message.getBytes());
```

Étape 2 : Configuration des confirmations de l'éditeur et de l'accusé de réception du client

Le processus de confirmation de l'envoi d'un message au courtier est appelé confirmation de l'éditeur. L'éditeur confirme que votre application est informée lorsque les messages ont été stockés de manière fiable. Les confirmations de l'éditeur peuvent également aider à contrôler le taux de messages stockés auprès du courtier. Sans la confirmation de l'éditeur, il n'y a aucune confirmation qu'un message a été traité correctement, et votre courtier peut supprimer les messages qu'il ne peut pas traiter.

De même, lorsqu'une application cliente envoie une confirmation de livraison et de consommation de messages au courtier, on parle d'accusé de réception pour le consommateur. La confirmation et l'accusé de réception sont essentiels pour garantir la sécurité des données lorsque vous travaillez avec des courtiers RabbitMQ.

L'accusé de réception de livraison du consommateur est généralement configuré sur l'application client. Lorsque vous travaillez avec AMQP 0-9-1, l'accusé de réception peut être activé en configurant la méthode. `basic.consume` Les clients AMQP 0-9-1 peuvent également configurer les confirmations de l'éditeur en envoyant la méthode. `confirm.select`

En règle générale, l'accusé de réception de livraison est activé dans un canal. Par exemple, lorsque vous travaillez avec la bibliothèque client Java RabbitMQ, vous pouvez utiliser l'`Channel#basicAck` pour mettre en place une confirmation positive `basic.ack` comme illustré dans l'exemple suivant.

```
// 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**
>
> Les messages sans accusé de réception doivent être mis en cache en mémoire. Vous pouvez limiter le nombre de messages qu'un consommateur récupère à l'avance en configurant les paramètres de pré-extraction pour une application client.

Vous pouvez configurer `consumer_timeout` pour détecter les cas où les consommateurs n'accusent pas réception des livraisons. Si le consommateur n'envoie pas d'accusé de réception dans le délai imparti, le canal sera fermé et vous recevrez un. `PRECONDITION_FAILED` Pour diagnostiquer l'erreur, utilisez l'UpdateConfigurationAPI pour augmenter la `consumer_timeout` valeur.

Étape 3 : Réduisez les files d'attente

Dans les déploiements en cluster, les files d'attente comportant un grand nombre de messages peuvent entraîner une surexploitation des ressources. Lorsqu'un agent est surutilisé, le redémarrage d'un agent Amazon MQ for RabbitMQ peut entraîner une dégradation supplémentaire des performances. En cas de redémarrage, les agents surexploités risquent de ne plus répondre dans l'état `REBOOT_IN_PROGRESS`.

Durant les fenêtres de maintenance, Amazon MQ effectue tous les travaux de maintenance un nœud à la fois pour s'assurer que l'agent reste opérationnel. Par conséquent, les files d'attente peuvent devoir se synchroniser à mesure que chaque nœud reprend l'opération. Pendant la synchronisation, les messages qui doivent être répliqués en miroirs sont chargés en mémoire à partir du volume Amazon Elastic Block Store (Amazon EBS) correspondant à traiter par lots. Le traitement des messages par lots permet aux files d'attente de se synchroniser plus rapidement.

Si les files d'attente sont courtes et que les messages sont petits, les files d'attente se synchronisent et reprennent le fonctionnement comme prévu. Toutefois, si la quantité de données dans un lot approche de la limite de mémoire du nœud, le nœud déclenche une alarme de mémoire élevée, mettant en pause la synchronisation de la file d'attente. Vous pouvez confirmer l'utilisation de la mémoire en comparant les métriques du nœud `RabbitMemUsed` et du nœud `RabbitMqMemLimit` broker dans CloudWatch. La synchronisation ne peut pas se terminer tant que les messages ne sont pas consommés ou supprimés, ou que le nombre de messages dans le lot est réduit.

Si la synchronisation des files d'attente est interrompue pour un déploiement en cluster, nous vous recommandons de consommer ou de supprimer des messages afin de réduire le nombre de messages dans les files d'attente. Une fois la profondeur de la file d'attente réduite et la synchronisation de la file d'attente terminée, l'état de l'agent passe à `RUNNING`. Pour résoudre une synchronisation de file d'attente interrompue, vous pouvez également appliquer une politique pour réduire la taille du lot de synchronisation des files d'attente.

Vous pouvez également définir des politiques de suppression automatique et de TTL afin de réduire de manière proactive l'utilisation des ressources et NACKs de limiter au maximum la communication avec les consommateurs. La mise en file d'attente de messages sur le courtier consomme beaucoup de ressources processeur, de sorte qu'un nombre élevé de messages peut affecter les performances du NACKs courtier.

Meilleures pratiques en matière d'optimisation des performances et d'efficacité dans Amazon MQ pour RabbitMQ

Vous pouvez optimiser les performances de votre Amazon MQ pour les courtiers RabbitMQ en maximisant le débit, en minimisant la latence et en garantissant une utilisation efficace des ressources. Suivez les meilleures pratiques suivantes pour optimiser les performances de votre application.

Étape 1 : maintenez la taille des messages en dessous de 1 Mo

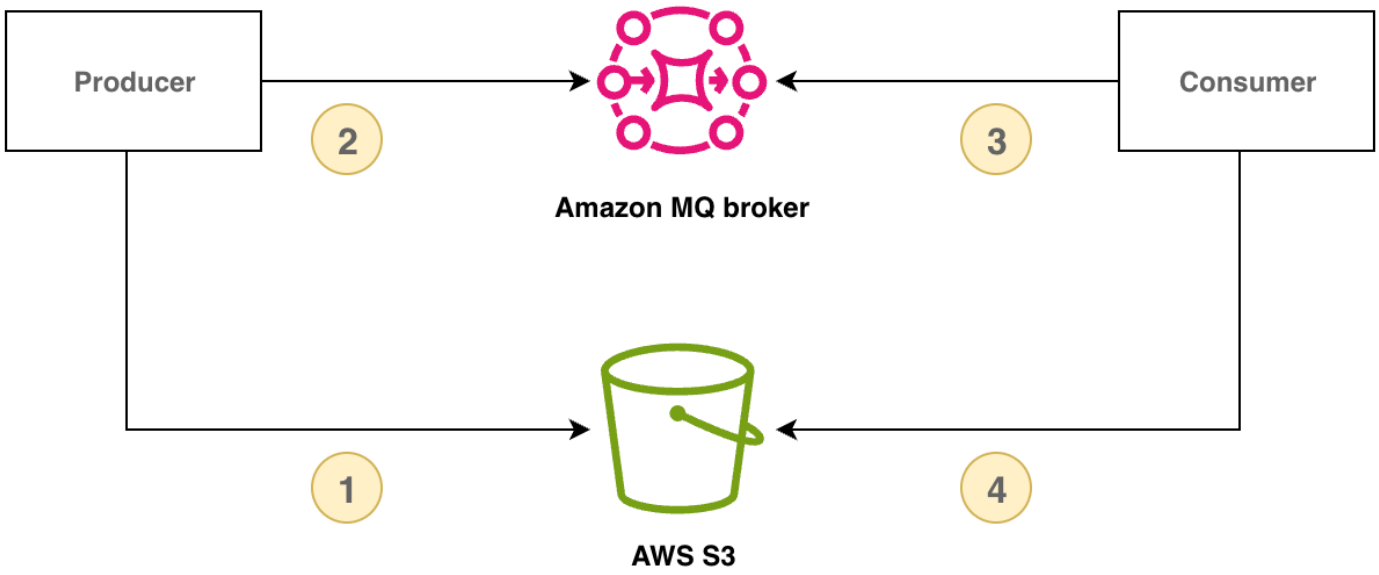
Nous recommandons de conserver les messages à moins de 1 mégaoctet (Mo) pour des performances et une fiabilité optimales.

RabbitMQ 3.13 prend en charge des tailles de message allant jusqu'à 128 Mo par défaut, mais les messages volumineux peuvent déclencher des alarmes de mémoire imprévisibles qui bloquent la publication et peuvent créer une pression mémoire élevée lors de la réplication des messages sur les nœuds. Les messages surdimensionnés peuvent également affecter les processus de redémarrage et de restauration des courtiers, ce qui augmente les risques pour la continuité du service et peut entraîner une dégradation des performances.

Stockez et récupérez des charges utiles importantes à l'aide du modèle de vérification des sinistres

Pour gérer les messages volumineux, vous pouvez implémenter le modèle de vérification des réclamations en stockant la charge utile du message dans un stockage externe et en envoyant uniquement l'identifiant de référence de la charge utile via RabbitMQ. Le consommateur utilise l'identifiant de référence de charge utile pour récupérer et traiter le message volumineux.

Le schéma suivant montre comment utiliser Amazon MQ pour RabbitMQ et Amazon S3 pour implémenter le modèle de vérification des réclamations.



L'exemple suivant illustre ce modèle à l'aide d'Amazon MQ, du AWS SDK pour Java 2.x et d'Amazon S3 :

1. Définissez d'abord une classe de message qui contiendra l'identifiant de référence Amazon S3.

```
class Message {
    // Other data fields of the message...

    public String s3Key;
    public String s3Bucket;
}
```

2. Créez une méthode d'éditeur qui stocke la charge utile dans Amazon S3 et envoie un message de référence via 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. Implémentez une méthode consommateur qui récupère la charge utile d'Amazon S3, la traite et supprime l'objet 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());
}
```

Étape 2 : Utilisation **`basic.consume`** et longue durée de vie des consommateurs

L'utilisation `basic.consume` auprès d'un consommateur de longue date est plus efficace que le sondage pour des messages individuels. `basic.get` Pour plus d'informations, consultez la section Sondage de messages individuels.

Étape 3 : Configuration de la pré-extraction

Vous pouvez utiliser la valeur de pré-extraction RabbitMQ pour optimiser la façon dont vos consommateurs consomment les messages. RabbitMQ implémente le mécanisme de pré-extraction des canaux fourni par AMQP 0-9-1 en appliquant le nombre de pré-extraction aux consommateurs plutôt qu'aux canaux. La valeur de pré-extraction est utilisée pour spécifier le nombre de messages envoyés au consommateur à un moment donné. Par défaut, RabbitMQ définit une taille de tampon illimitée pour les applications client.

Il existe une variété de facteurs à prendre en compte lors de la définition d'un nombre de pré-extraction pour vos consommateurs RabbitMQ. Tout d'abord, considérez l'environnement et la

configuration de vos clients. Étant donné que les consommateurs doivent conserver tous les messages en mémoire au fur et à mesure qu'ils sont traités, une valeur de pré-extraction élevée peut avoir un impact négatif sur les performances de vos consommateurs et, dans certains cas, peut entraîner un blocage potentiel d'un consommateur. De même, l'agent RabbitMQ conserve lui-même tous les messages qu'il envoie mis en mémoire cache jusqu'à ce qu'il reçoive l'accusé de réception du consommateur. Une valeur de pré-extraction élevée peut entraîner une perte de mémoire rapide de votre serveur RabbitMQ si l'accusé de réception automatique n'est pas configuré pour les consommateurs et si les consommateurs prennent un temps relativement long pour traiter les messages.

En prenant en compte les considérations ci-dessus, nous vous recommandons de toujours définir une valeur de pré-extraction afin d'éviter les situations où un agent RabbitMQ ou ses consommateurs manquent de mémoire en raison d'un grand nombre de messages non traités ou sans accusés de réception. Si vous avez besoin d'optimiser vos agents pour traiter de grands volumes de messages, vous pouvez tester vos agents et vos consommateurs à l'aide d'une plage de comptes de pré-extraction afin de déterminer la valeur à laquelle les frais généraux du réseau deviennent largement insignifiants par rapport au temps nécessaire au traitement des messages par un consommateur.

> **Note**
>
> - Si vos applications client ont été configurées pour reconnaître automatiquement la remise des messages aux consommateurs, la définition d'une valeur de pré-extraction n'aura aucun effet.
> - Tous les messages pré-extraits sont supprimés de la file d'attente.

L'exemple suivant montre la définition d'une valeur de pré-extraction de `10` pour un seul consommateur utilisant la bibliothèque 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**
>
> Dans la bibliothèque client Java RabbitMQ, la valeur par défaut de la propriété `globalest` définie sur `false`, donc l'exemple ci-dessus peut être écrit simplement comme `channel.basicQos(10)`.

Étape 4 : Utiliser Celery 5.5 ou version ultérieure avec les files d'attente du quorum

Python Celery, un système de file d'attente de tâches distribué, peut générer de nombreux messages non critiques en cas de charge de tâches élevée. Cette activité supplémentaire du courtier peut déclencher the section called “ `RABBITMQ_MEMORY_ALARM` ” et entraîner une indisponibilité du courtier. Pour réduire le risque de déclenchement d'une alarme mémoire, procédez comme suit :

Pour toutes les versions de Celery

1. Désactivez-le `task_create_missing_queues`pour réduire le taux de désabonnement des files d'attente.
2. Ensuite, désactivez-le `worker_enable_remote_control` pour arrêter la création dynamique de `celery@...pidbox` files d'attente. Cela réduira le taux de désabonnement des files d'attente chez le courtier.

   ```
   worker_enable_remote_control = false
   ```
3. Pour réduire davantage l'activité des messages non critiques, désactivez Celery worker-send-task-eventsen ne les incluant pas `-E` ou en les `--task-events` signalant au démarrage de votre application Celery.
4. Démarrez votre application Celery en utilisant les paramètres suivants :

   ```
   celery -A app_name worker --without-heartbeat --without-gossip --without-mingle
   ```

Pour les versions 5.5 et supérieures de Celery

1. Passez à la version 5.5 de Celery, la version minimale qui prend en charge les files d'attente de quorum, ou à une version ultérieure. Pour vérifier quelle version de Celery vous utilisez, utilisez`celery --version`. Pour plus d'informations sur les files d'attente pour le quorum, consultezthe section called “files d'attente pour le quorum”.

2. Après la mise à niveau vers Celery 5.5 ou version ultérieure, configurez `task_default_queue_type` sur « quorum ».
3. Ensuite, vous devez également activer Publier les confirmations dans les options de transport des courtiers :

```
broker_transport_options = {"confirm_publish": True}
```

Meilleures pratiques en matière de résilience et de surveillance du réseau dans Amazon MQ pour RabbitMQ

La résilience du réseau et la surveillance des métriques des courtiers sont essentielles pour garantir la fiabilité des applications de messagerie. Suivez les meilleures pratiques suivantes pour mettre en œuvre des mécanismes de restauration automatique et des stratégies de surveillance des ressources.

Étape 1 : restauration automatique en cas de défaillance du réseau

Nous vous recommandons de toujours activer la récupération automatique du réseau pour éviter les temps d'arrêt importants en cas d'échec des connexions client aux nœuds RabbitMQ. La bibliothèque client Java RabbitMQ prend en charge la récupération automatique du réseau par défaut, en commençant par la version `4.0.0`.

La restauration automatique de la connexion est déclenchée si une exception non gérée est émise dans la I/O boucle de connexion, si le délai d'expiration d'une opération de lecture du socket est détecté ou si le serveur rate un battement de cœur.

Dans les cas où la connexion initiale entre un client et un nœud RabbitMQ échoue, la récupération automatique ne sera pas déclenchée. Nous vous recommandons d'écrire votre code d'application pour tenir compte des échecs de connexion initiaux en tentant de nouveau la connexion. L'exemple suivant illustre la nouvelle tentative d'échec réseau initial à l'aide de la bibliothèque 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**
>
> Si une application ferme une connexion à l'aide de la méthode `Connection.Close`, la récupération automatique du réseau ne sera ni activée ni déclenchée.

Étape 2 : Surveiller les métriques et les alarmes des courtiers

Nous vous recommandons de surveiller régulièrement CloudWatch les métriques et les alarmes de votre courtier Amazon MQ pour RabbitMQ afin d'identifier et de résoudre les problèmes potentiels avant qu'ils n'affectent votre application de messagerie. La surveillance proactive est essentielle pour maintenir une application de messagerie résiliente et garantir des performances optimales.

Amazon MQ pour RabbitMQ publie des statistiques CloudWatch qui fournissent des informations sur les performances des courtiers, l'utilisation des ressources et le flux de messages. Les indicateurs clés à surveiller incluent l'utilisation de la mémoire et l'utilisation du disque. Vous pouvez configurer des CloudWatch alarmes lorsque votre courtier approche des limites de ressources ou subit une dégradation des performances.

Surveillez les indicateurs essentiels suivants :

`RabbitMQMemUsed` et **`RabbitMQMemLimit`**

Surveillez l'utilisation de la mémoire pour éviter les alarmes susceptibles de bloquer la publication des messages.

`RabbitMQDiskFree` et **`RabbitMQDiskFreeLimit`**

Surveillez l'utilisation du disque pour éviter les problèmes d'espace disque susceptibles de provoquer des défaillances du broker.

Pour les déploiements de clusters, surveillez également les métriques spécifiques aux nœuds afin d'identifier les problèmes spécifiques aux nœuds.

> **Note**
>
> Pour plus d'informations sur la façon de prévenir une alarme de mémoire trop importante, voir Corriger et empêcher une alarme de mémoire trop importante.

Didacticiels RabbitMQ

Les didacticiels suivants vous montrent comment configurer et utiliser RabbitMQ sur Amazon MQ. Pour en savoir plus sur l'utilisation des bibliothèques client prises en charge dans une variété de langages de programmation tels que Node.js, Python, .NET, etc., consultez Didacticiels RabbitMQ du Guide de démarrage RabbitMQ.

Rubriques

Modification des préférences d'agent

Vous pouvez modifier les préférences de votre courtier, par exemple en activant ou en désactivant CloudWatch les journaux à l'aide du AWS Management Console.

Modifier les options de l'agent RabbitMQ

1. Connectez-vous à la console Amazon MQ.
2. Dans la liste des courtiers, sélectionnez votre courtier (par exemple MyBroker), puis choisissez Modifier.
3. Sur la *MyBroker* page Modifier, dans la section Spécifications, sélectionnez une version du moteur Broker ou un type d'instance Broker.
4. Dans la section CloudWatch Journaux, cliquez sur le bouton bascule pour activer ou désactiver les journaux généraux. Les autres étapes sont requises.

   > **Note**
   >
   > - Pour les courtiers RabbitMQ, Amazon MQ utilise automatiquement un rôle lié au service (SLR) pour publier des journaux généraux sur. CloudWatch Pour de plus amples informations, consultez the section called “Utilisation des rôles liés à un service”.
   > - Amazon MQ ne prend pas en charge la journalisation d'audit pour les agents RabbitMQ.

5. Dans la section Maintenance, configurez le programme de maintenance de votre agent :

   Pour mettre à niveau le broker vers de nouvelles versions au fur et à mesure de sa AWS publication, choisissez Activer les mises à niveau automatiques des versions mineures. Les mises à niveau automatiques se produisent pendant la fenêtre de maintenance définie par le jour de la semaine, l'heure de la journée (au format 24 heures) et le fuseau horaire (UTC par défaut).
6. Choisissez Schedule Modifications (Planifier les modifications).

   > **Note**
   >
   > Si vous choisissez uniquement Enable automatic minor version upgrades (Activer les mises à niveau automatiques des versions mineures), le bouton est remplacé par Save (Enregistrer), car aucun redémarrage d'agent n'est nécessaire.

   Vos préférences sont appliquées à votre agent à l'heure spécifiée.

Utilisation de Python Pika avec Amazon MQ pour RabbitMQ

Le didacticiel suivant montre comment vous pouvez configurer un client Pika Python avec TLS configuré pour se connecter à un agent Amazon MQ pour RabbitMQ. Pika est une implémentation Python du protocole AMQP 0-9-1 pour RabbitMQ. Ce didacticiel vous explique comment installer Pika, déclarer une file d'attente, configurer un éditeur pour envoyer des messages à la plateforme d'échange par défaut du courtier et configurer un consommateur pour recevoir les messages de la file d'attente.

Rubriques

Conditions préalables

Pour compléter les cinq premières étapes de ce didacticiel, vous devez disposer des éléments suivants :

- Un agent Amazon MQ pour RabbitMQ. Pour de plus amples informations, consultez l'article sur la création d'un agent Amazon MQ pour RabbitMQ.
- L'outil Python 3 correspondant à votre système d'exploitation.
- L'outil Pika installé à l'aide de Python `pip`. Pour installer Pika, ouvrez une nouvelle fenêtre dans le terminal et exécutez la procédure suivante.

```
$ python3 -m pip install pika
```

Permissions

Pour pouvoir suivre ce didacticiel, vous aurez besoin d'au moins un utilisateur d'agent Amazon MQ pour RabbitMQ autorisé à écrire et à lire depuis un vhost. Le tableau suivant décrit les autorisations minimales nécessaires sous forme de modèles d'expressions régulières (regexp).

Étiquettes	Configurer regexp	Regexp en écriture	Lire regexp
none		`.*`	`.*`

Les autorisations utilisateur répertoriées accordent uniquement des autorisations en lecture et en écriture et ne donnent aucun accès au plugin de gestion permettant d'effectuer des opérations administratives sur l'agent. Vous pouvez encore restreindre davantage les autorisations en fournissant des modèles de regexp qui limitent l'accès de l'utilisateur à des files d'attente spécifiées. Par exemple, si vous remplacez le modèle de regexp en lecture par `^[hello world].*`, l'utilisateur sera uniquement autorisé à lire des files d'attente commençant par `hello world`.

Pour plus d'informations sur la création d'utilisateurs RabbitMQ et la gestion de balises et autorisations utilisateur, consultez Amazon MQ pour les utilisateurs du broker RabbitMQ.

Première étape : création d'un client Python Pika de base

Pour créer une classe de client de base Python Pika capable de définir un constructeur et de fournit le contexte SSL nécessaire à la configuration TLS lors de l'interaction avec un agent Amazon MQ pour RabbitMQ, procédez comme suit.

1. Ouvrez une nouvelle fenêtre dans le terminal, créez un nouveau répertoire pour votre projet et accédez-y.

   ```
   $ mkdir pika-tutorial
   $ cd pika-tutorial
   ```

2. Créez un fichier nommé `basicClient.py` contenant le code suivant.

   ```
   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()
```

Vous pouvez désormais définir des classes supplémentaires pour votre éditeur et votre consommateur qui héritent de `BasicPikaClient`.

Deuxième étape : création d'un éditeur et envoi d'un message

Pour créer un éditeur capable de déclarer une file d'attente et d'envoyer un message unique, procédez comme suit.

1. Copiez le contenu de l'exemple de code suivant et enregistrez-le localement sous `publisher.py` dans le même répertoire que celui que vous avez créé à l'étape précédente.

```
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 classe `BasicMessageSender` hérite de `BasicPikaClient` et implémente des méthodes supplémentaires pour déclarer une file d'attente, envoyer un message sur celle-ci et fermer les connexions. L'exemple de code achemine un message vers l'échange par défaut, avec une clé de routage identique au nom de la file d'attente.

2. Sous `if __name__ == "__main__":`, remplacez les paramètres transmis à la déclaration de constructeur `BasicMessageSender` par les informations suivantes.

   - **<broker-id>** – ID unique généré par Amazon MQ pour l'agent. Vous pouvez analyser l'ID de votre ARN d'agent. Par exemple, avec l'ARN suivant, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, l'ID de l'agent serait `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
   - **<username>** : nom d'utilisateur d'agent disposant des autorisations suffisantes pour écrire des messages à l'agent.
   - **<password>** : mot de passe d'un utilisateur d'agent disposant des autorisations suffisantes pour écrire des messages à l'agent.

- **<region>**— La AWS région dans laquelle vous avez créé votre courtier Amazon MQ pour RabbitMQ. Par exemple, `us-west-2`.

3. Exécutez la commande suivante dans le répertoire où vous avez créé `publisher.py`.

```
$ python3 publisher.py
```

Si le code s'exécute correctement, vous verrez la sortie suivante dans la fenêtre de votre terminal.

```
Trying to declare queue(hello world queue)...
Sent message. Exchange: , Routing Key: hello world queue, Body: b'Hello World!'
```

Troisième étape : créer un consommateur et recevoir un message

Pour créer un consommateur qui reçoit un seul message de la file d'attente, procédez comme suit.

1. Copiez le contenu de l'exemple de code suivant et enregistrez-le sous `consumer.py` dans le même répertoire.

```
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()
```

Semblable à l'éditeur que vous avez créé à l'étape précédente, `BasicMessageReceiver` hérite de méthodes supplémentaires `BasicPikaClient` et implémente des méthodes supplémentaires pour recevoir un seul message et fermer les connexions.

2. Sous la déclaration `if __name__ == "__main__":`, remplacez les paramètres transmis au constructeur `BasicMessageReceiver` par vos données.
3. Exécutez la commande suivante dans le répertoire de votre projet.

```
$ python3 consumer.py
```

Si le code s'exécute correctement, le corps du message et les en-têtes, y compris la clé de routage, s'affichent dans la fenêtre de votre terminal.

```
<Basic.GetOk(['delivery_tag=1', 'exchange=', 'message_count=0',
 'redelivered=False', 'routing_key=hello world queue'])> <BasicProperties> b'Hello
 World!'
```

Étape quatre : (facultatif) configuration d'une boucle d'événements et consommation des messages

Pour consommer plusieurs messages d'une file d'attente, utilisez la méthode `basic_consume` de Pika ainsi qu'une fonction de rappel, comme illustré ci-dessous.

1. Dans `consumer.py`, ajoutez la définition de méthode suivante à la 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. Dans `consumer.py`, sous `if __name__ == "__main__":`, invoquez la méthode `consume_messages` que vous avez définie à l'étape précédente.

```
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. Exécutez `consumer.py` à nouveau. Si tout se passe normalement, les messages mis en file d'attente s'afficheront dans la fenêtre de votre terminal.

```
[*] Waiting for messages. To exit press CTRL+C
[x] Received b'Hello World!'
[x] Received b'Hello World!'
...
```

Quelle est la prochaine étape ?

- Pour en savoir plus sur les autres bibliothèques de clients RabbitMQ prises en charge, consultez la documentation sur les clients RabbitMQ disponible sur le site web de RabbitMQ.

Résolution de la synchronisation de la file d'attente mise en pause RabbitMQ

Dans un déploiement en cluster Amazon MQ for RabbitMQ, les messages publiés dans chaque file d'attente sont répliqués sur trois nœuds d'agent. Cette réplication, appelée mise en miroir, fournit une haute disponibilité (HA) pour les agents RabbitMQ. Les files d'attente d'un déploiement en cluster se composent d'un réplica principal sur un nœud et un ou plusieurs miroirs. Chaque opération appliquée à une file d'attente mise en miroir, y compris les messages de mise en file d'attente, est d'abord appliquée à la file d'attente principale, puis répliquée sur ses miroirs.

Par exemple, considérez une file d'attente en miroir répliquée sur trois nœuds : le nœud principal (`main`) et deux miroirs (`mirror-1` et `mirror-2`). Si tous les messages de cette file d'attente en miroir sont propagés avec succès à tous les miroirs, la file d'attente est synchronisée. Si un nœud (`mirror-1`) devient indisponible pendant un intervalle de temps, la file d'attente est toujours opérationnelle et peut continuer à mettre en file d'attente des messages. Toutefois, pour que la file d'attente se synchronise, les messages publiés dans `main` tandis que `mirror-1` n'est pas disponible doivent être répliqués dans `mirror-1`.

Pour plus d'informations sur la mise en miroir, consultez Files d'attente mises en miroir classiques sur le site web RabbitMQ.

Maintenance et synchronisation des files d'attente

Lors de fenêtres de maintenance, Amazon MQ effectue tous les travaux de maintenance un nœud à la fois pour s'assurer que l'agent reste opérationnel. Par conséquent, les files d'attente peuvent devoir se synchroniser à mesure que chaque nœud reprend l'opération. Pendant la synchronisation, les messages qui doivent être répliqués en miroirs sont chargés en mémoire à partir du volume Amazon Elastic Block Store (Amazon EBS) correspondant à traiter par lots. Le traitement des messages par lots permet aux files d'attente de se synchroniser plus rapidement.

Si les files d'attente sont courtes et que les messages sont petits, les files d'attente se synchronisent et reprennent le fonctionnement comme prévu. Toutefois, si la quantité de données dans un lot approche de la limite de mémoire du nœud, le nœud déclenche une alarme de mémoire élevée,

mettant en pause la synchronisation de la file d'attente. Vous pouvez confirmer l'utilisation de la mémoire en comparant les métriques du nœud `RabbitMemUsed` et du nœud `RabbitMqMemLimit` broker dans CloudWatch. La synchronisation ne peut pas se terminer tant que les messages ne sont pas consommés ou supprimés, ou que le nombre de messages dans le lot est réduit.

> **Note**
>
> La réduction de la taille du lot de synchronisation des files d'attente peut entraîner un plus grand nombre de transactions de réplication.

Pour résoudre une synchronisation de file d'attente mise en pause, suivez les étapes de ce didacticiel, qui illustre l'application d'une politique `ha-sync-batch-size` et le redémarrage de la synchronisation de la file d'attente.

Rubriques

Conditions préalables

Pour ce didacticiel, vous devez disposer d'un utilisateur d'agent Amazon MQ for RabbitMQ avec les autorisations d'administrateur. Vous pouvez utiliser l'utilisateur administrateur créé lors de la création de l'agent pour la première fois, ou un autre utilisateur que vous avez créé par la suite. Le tableau suivant fournit la balise utilisateur administrateur et les autorisations nécessaires en tant que modèles d'expression régulière (regexp).

Étiquettes	Lire regexp	Configurer regexp	Regexp en écriture
`administrator`	`.*`	`.*`	`.*`

Pour plus d'informations sur la création d'utilisateurs RabbitMQ et la gestion de balises et autorisations utilisateur, consultez Amazon MQ pour les utilisateurs du broker RabbitMQ.

Étape 1 : Appliquer une politique **ha-sync-batch-size**

Les procédures suivantes illustrent l'ajout d'une politique qui s'applique à toutes les files d'attente créées sur l'agent. Vous pouvez utiliser la console web RabbitMQ ou l'API de gestion RabbitMQ. Pour plus d'informations, consultez Plugin de gestion sur le site web RabbitMQ.

Pour appliquer une politique **ha-sync-batch-size** à l'aide de la console web RabbitMQ

1. Connectez-vous à la console Amazon MQ.
2. Dans le panneau de navigation de gauche, choisissez Brokers (Agents).
3. Dans la liste des agents, choisissez le nom de l'agent auquel vous souhaitez appliquer la nouvelle politique.
4. Sur la page de l'agent, dans la section Connection (Connexions), choisissez l'URL RabbitMQ web console (Console web RabbitMQ). La console web RabbitMQ s'ouvre dans un nouvel onglet ou une nouvelle fenêtre du navigateur.
5. Connectez-vous à la console Web RabbitMQ à l'aide de vos informations d'identification de connexion d'administrateur d'agent.
6. Dans la console web RabbitMQ, en haut de la page, choisissez Admin (Administrateur).
7. Dans la page Admin (Administrateur), dans le volet de navigation droit, choisissez Policies (Politiques).
8. Dans la page Policies (Politiques), vous pouvez consulter la liste actuelle User policies (Politiques utilisateur) de l'agent. Sous User policies (Politiques utilisateur), développez Add/update a policy (Ajouter ou mettre à jour une politique).

   > **Note**
   >
   > Par défaut, les clusters Amazon MQ for RabbitMQ sont créés avec une politique d'agent initiale nommée `ha-all-AWS-OWNED-DO-NOT-DELETE`. Amazon MQ gère cette politique pour s'assurer que chaque file d'attente de l'agent est répliquée sur les trois nœuds et que les files d'attente sont synchronisées automatiquement.

9. Pour créer une politique d'agent, sous Add/update a policy (Ajouter ou mettre à jour une politique), procédez comme suit :

   a. Dans Name (Name), attribuez un nom à votre politique IAM, par exemple **batch-size-policy**.

b. Pour Pattern (Modèle), entrez le modèle regexp **`.*`** afin que la politique corresponde à toutes les files d'attente sur l'agent.

c. Pour Apply to (Appliquer à), choisissez Exchanges and queues (Échanges et files d'attente) dans la liste déroulante.

d. Pour Priority (Priorité), entrez un entier supérieur à toutes les autres politiques appliquées au vhost. Vous pouvez appliquer exactement un ensemble de définitions de politiques aux files d'attente et aux échanges RabbitMQ à tout moment. RabbitMQ choisit la politique correspondante avec la valeur de priorité la plus élevée. Pour plus d'informations sur les priorités de politique et sur la façon de combiner des politiques, veuillez consulter Policies (Politiques) dans la documentation du serveur RabbitMQ.

e. Pour Definition (Définition), ajoutez les paires clé-valeur suivantes :

- **`ha-sync-batch-size=`***`100`*. Choisissez Numéro dans la liste déroulante.

> **Note**
>
> Vous devrez peut-être ajuster et calibrer la valeur de `ha-sync-batch-size` en fonction du nombre et de la taille des messages non synchronisés dans vos files d'attente.

- **`ha-mode=all`**. Choisissez String (Chaîne) dans la liste déroulante.

> **Important**
>
> La définition `ha-mode` est requise pour toutes les politiques relatives à la haute disponibilité. Son omission entraîne un échec de validation.

- **`ha-sync-mode=automatic`**. Choisissez String (Chaîne) dans la liste déroulante.

> **Note**
>
> La définition `ha-sync-mode` est requise pour toutes les politiques relatives à la haute disponibilité. Si elle est omise, Amazon MQ ajoute automatiquement la définition.

f. Choisissez Add / update policy (Ajouter/mettre à jour une politique).

10. Confirmez que la nouvelle politique apparaît dans la liste User policies (Politiques utilisateur).

Pour appliquer une politique **`ha-sync-batch-size`** à l'aide de l'API de gestion RabbitMQ

1. Connectez-vous à la console Amazon MQ.
2. Dans le panneau de navigation de gauche, choisissez Brokers (Agents).
3. Dans la liste des agents, choisissez le nom de l'agent auquel vous souhaitez appliquer la nouvelle politique.
4. Sur la page de l'agent, dans la section Connection (Connexion), notez l'URL de la console web RabbitMQ. Il s'agit du point de terminaison de l'agent que vous utilisez dans une requête HTTP.
5. Ouvrez une nouvelle fenêtre de terminal ou de ligne de commande.
6. Pour créer une politique d'agent, entrez la commande `curl` suivante. Cette commande suppose la présence d'une file d'attente sur le vhost `/` par défaut, qui est encodé en tant que `%2F`.

   > **Note**
   >
   > Remplacez *`username`* et par *`password`* les informations de connexion de l'administrateur de votre courtier. Vous devrez peut-être ajuster et calibrer la valeur de `ha-sync-batch-size` (*`100`*) en fonction du nombre et de la taille des messages non synchronisés dans vos files d'attente. Remplacez le point de terminaison de l'agent par l'URL que vous avez notée précédemment.

   ```
   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. Pour confirmer que la nouvelle politique est ajoutée aux politiques utilisateur de votre agent, entrez la commande `curl` suivante pour répertorier toutes les politiques de l'agent.

   ```
   curl -i -u username:password https://b-589c045f-f8ln-4ab0-a89c-co62e1c32ef8.mq.us-
   west-2.amazonaws.com/api/policies
   ```

Étape 2 : Redémarrer la synchronisation de la file d'attente

Après avoir appliqué une nouvelle politique `ha-sync-batch-size` à votre agent, redémarrez la synchronisation de la file d'attente.

Pour redémarrer la synchronisation de la file d'attente à l'aide de la console web RabbitMQ

> **Note**
>
> Pour ouvrir la console web RabbitMQ, reportez-vous aux instructions précédentes à l'étape 1 de ce didacticiel.

1. Dans la console web RabbitMQ, en haut de la page, choisissez Queues (Fils d'attent).
2. Dans la page Files d'attente (Queues), sous All queues (Toutes les files d'attente), localisez votre file d'attente mise en pause. Dans la ligne Politique, votre file d'attente doit indiquer le nom de la nouvelle politique que vous avez créée (par exemple,`batch-size-policy`).
3. Pour redémarrer le processus de synchronisation avec une taille de lot réduite, annulez d'abord la synchronisation de la file d'attente. Redémarrez ensuite la synchronisation des files d'attente.

> **Note**
>
> Si la synchronisation s'interrompt et ne se termine pas correctement, essayez de réduire la valeur `ha-sync-batch-size` et de redémarrer la synchronisation de la file d'attente.

Étapes suivantes

- Une fois que votre file d'attente est correctement synchronisée, vous pouvez surveiller la quantité de mémoire utilisée par vos nœuds RabbitMQ en consultant la métrique Amazon. CloudWatch `RabbitMQMemUsed` Vous pouvez également afficher la métrique `RabbitMQMemLimit` pour surveiller la limite de mémoire d'un nœud. Pour plus d’informations, consultez Accès aux CloudWatch métriques pour Amazon MQ et CloudWatch Mesures disponibles pour Amazon MQ pour les courtiers RabbitMQ.
- Pour empêcher la synchronisation des files d'attente interrompues, nous vous recommandons de conserver les files d'attente courtes et de traiter les messages. Pour les applications ayant une taille de message plus grande, nous vous recommandons également de mettre à niveau

votre type d'instance d'agent vers une taille d'instance plus grande avec plus de mémoire. Pour plus d'informations sur les types d'instances de courtier et sur la modification des préférences du courtier, consultezModification des préférences d'agent.

- Lorsque vous créez un nouvel agent Amazon MQ for RabbitMQ, Amazon MQ applique un ensemble de politiques par défaut et de limites d'hôte virtuel pour optimiser les performances de l'agent. Si votre agent ne dispose pas des politiques et limites par défaut recommandées, nous vous recommandons de les créer vous-même. Pour plus d'informations sur la création de politiques par défaut et de limites de vhost, consultez https://docs.aws.amazon.com//amazon-mq/latest/developer-guide/rabbitmq-defaults.html.

Ressources connexes

- UpdateBrokerInput— Utilisez cette propriété de courtier pour mettre à jour un type d'instance de courtier à l'aide de l'API Amazon MQ.
- Paramètres et politiques (Documentation du serveur RabbitMQ) – Pour en savoir plus sur les paramètres et les politiques RabbitMQ, consultez le site web RabbitMQ.
- API HTTP de gestion RabbitMQ – En savoir plus sur l'API de gestion RabbitMQ.

Réduction du nombre de connexions et de canaux

Les connexions à votre courtier RabbitMQ sur Amazon MQ peuvent être fermées soit par vos applications clientes, soit en les fermant manuellement à l'aide de la console Web RabbitMQ. Pour fermer une connexion à l'aide de la console Web RabbitMQ, procédez comme suit :

1. Connectez-vous à la console AWS Management Console Web RabbitMQ de votre courtier et ouvrez-la.
2. Sur la console RabbitMQ, choisissez l’onglet Connexions.
3. Dans la page de Connexions, sous Toutes les connexions, choisissez le nom de la connexion à fermer dans la liste.
4. Sur la page des informations de la connexion, choisissez Fermez cette connexion pour développer la section, choisissez Forcer la fermeture. Le cas échéant, vous pouvez remplacer le texte par défaut de Raison avec votre propre description. RabbitMQ sur Amazon MQ renverra la raison que vous avez spécifiée au client lorsque vous fermez la connexion.
5. Choisissez OK dans la boîte de dialogue pour confirmer et fermer la connexion.

Lorsque vous fermez une connexion, tous les canaux associés à une connexion fermée seront également fermés.

> **Note**
>
> Vos applications clientes peuvent être configurées pour rétablir automatiquement les connexions avec l'agent après leur fermeture. Dans ce cas, la fermeture des connexions à partir de la console Web Broker ne suffira pas à réduire le nombre de connexions ou de canaux.

Pour les agents sans accès public, vous pouvez bloquer temporairement les connexions en refusant le trafic entrant sur le port de protocole de message approprié, par exemple le port 5671 pour les connexions AMQP. Vous pouvez bloquer le port du groupe de sécurité que vous avez fourni à Amazon MQ lors de la création de l'agent. Pour de plus amples informations sur la modification d'un groupe de sécurité de VPC, veuillez consulter Security Groups for Your VPC (Groupes de sécurité pour votre VPC) dans le Guide de l'utilisateur .

Étape 2 : Connectez une application basée sur JVM à votre courtier

Après avoir créé un agent RabbitMQ, vous pouvez y connecter votre application. Les exemples suivants montrent comment utiliser la bibliothèque client Java RabbitMQ pour créer une connexion à votre agent, créer une file d'attente et envoyer un message. Vous pouvez vous connecter à des agents RabbitMQ à l'aide des bibliothèques client RabbitMQ prises en charge pour une variété de langages. Pour plus d'informations sur les bibliothèques clientes RabbitMQ prises en charge, voir Bibliothèques clientes RabbitMQ et outils de développement.

Conditions préalables

> **Note**
>
> Les étapes préalables suivantes ne s'appliquent qu'aux agents RabbitMQ créés sans accès public. Si vous créez un agent avec accès public, vous pouvez les ignorer.

Activer les attributs du VPC

Pour vous assurer que votre agent est accessible dans votre VPC, vous devez activer les attributs `enableDnsHostnames` et `enableDnsSupport` du VPC. Pour plus d'informations, consultez Prise en charge du DNS dans votre VPC dans le Guide de l'utilisateur Amazon VPC.

Activer les connexions entrantes

1. Connectez-vous à la console Amazon MQ.
2. Dans la liste des courtiers, choisissez le nom de votre courtier (par exemple, MyBroker).
3. Sur la ***MyBroker***page, dans la section Connexions, notez les adresses et les ports de l'URL de la console Web du courtier et des protocoles au niveau du fil.
4. Dans la section Details (Détails), sous Security and network (Sécurité et réseau), choisissez le nom de votre groupe de sécurité ou

   .

   La page Groupes de sécurité du tableau de bord EC2 est affichée.
5. Dans la liste des groupes de sécurité, choisissez votre groupe de sécurité.
6. Au bas de la page, choisissez Entrant, puis Modifier.
7. Dans la boîte de dialogue Edit inbound rules (Modifier les règles entrantes), ajoutez une règle pour chaque URL ou point de terminaison pour qu'ils soient accessibles publiquement (l'exemple suivant montre comment procéder pour une console web d'agent).

   a. Choisissez Add Rule (Ajouter une règle).

   b. Pour Type, sélectionnez Custom TCP (TCP personnalisé).

   c. Pour Source, laissez l'option Custom (Personnalisée) sélectionnée, puis tapez l'adresse IP du système qui doit pouvoir accéder à la console web (par exemple, `192.0.2.1`).

   d. Choisissez Enregistrer.

      Votre agent peut désormais accepter les connexions entrantes.

Ajout de dépendances Java

Si vous utilisez Apache Maven pour automatiser les builds, ajoutez la dépendance suivante à votre fichier `pom.xml`. Pour plus d'informations sur les fichiers de modèle d'objet de projet dans Apache Maven, voir Présentation du POM.

```
<dependency>
    <groupId>com.rabbitmq</groupId>
    <artifactId>amqp-client</artifactId>
    <version>5.9.0</version>
</dependency>
```

Si vous utilisez Gradle pour automatiser les builds, déclarez la dépendance suivante.

```
dependencies {
    compile 'com.rabbitmq:amqp-client:5.9.0'
}
```

Importer les classe **`Connection`** et **`Channel`**

Le client Java RabbitMQ utilise `com.rabbitmq.client` comme paquet de premier niveau, avec les classes d'API `Connection` et `Channel` représentant une connexion et un canal AMQP 0-9-1, respectivement. Importez les classe `Connection` et `Channel` avant de les utiliser, comme illustré dans l'exemple suivant.

```
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.Channel;
```

Créer un **`ConnectionFactory`** et se connecter à votre agent

Utilisez l'exemple suivant pour créer une instance de la classe `ConnectionFactory` avec les paramètres donnés. Utilisation de la méthode `setHost` pour configurer le point de terminaison de l'agent noté précédemment. Pour les connexions au niveau filaire `AMQPS`, utilisez le port `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();
```

Publier un message dans un échange

Vous pouvez utiliser `Channel.basicPublish` pour publier des messages dans un échange. L'exemple suivant utilise la classe `Builder` AMQP pour créer un objet de propriétés de message avec le type de contenu `plain/text`.

```
byte[] messageBodyBytes = "Hello, world!".getBytes();
channel.basicPublish(exchangeName, routingKey,
             new AMQP.BasicProperties.Builder()
               .contentType("text/plain")
               .userId("userId")
               .build(),
               messageBodyBytes);
```

> **Note**
>
> Notez que `BasicProperties` est une classe interne de la classe de support générée automatiquement, `AMQP`.

S'abonner à une file d'attente et recevoir un message

Vous pouvez recevoir un message en vous abonnant à une file d'attente à l'aide de l'interface `Consumer`. Une fois abonné, les messages seront alors remis automatiquement dès leur arrivée.

La façon la plus simple d'implémenter un objet `Consumer` est d'utiliser la sous-classe `DefaultConsumer`. Un objet `DefaultConsumer` peut être transmis comme faisant partie d'un appel `basicConsume` pour configurer l'abonnement comme illustré dans l'exemple suivant.

```
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**
>
> Parce que nous avons spécifié `autoAck = false`, il est nécessaire d'accuser réception des messages remis à `Consumer`, le plus commodément fait dans la méthode `handleDelivery`, comme illustré dans l'exemple.

Fermer votre connexion et se déconnecter de l'agent

Afin de vous déconnecter de votre agent RabbitMQ, fermez à la fois le canal et la connexion comme indiqué ci-dessous.

```
channel.close();
conn.close();
```

> **Note**
>
> Pour plus d'informations sur l'utilisation de la bibliothèque cliente Java RabbitMQ, consultez le Guide de l'API client Java RabbitMQ.

Étape 3 : (Facultatif) Se connecter à une AWS Lambda fonction

AWS Lambda peut se connecter à votre courtier Amazon MQ et en consommer les messages. Lorsque vous connectez un agent à Lambda, vous créez un mappage de la source d'événement qui lit les messages d'une file d'attente et appelle la fonction de manière synchrone. Le mappage de la source d'événements que vous créez lit les messages de votre agent par lots et les convertit en une charge utile Lambda sous la forme d'un objet JSON.

Pour connecter votre agent à une fonction Lambda

1. Ajoutez les autorisations de rôle IAM suivantes au rôle d'exécution de votre fonction Lambda.

   - mq : DescribeBroker
   - EC2 : CreateNetworkInterface
   - EC2 : DeleteNetworkInterface
   - EC2 : DescribeNetworkInterfaces
   - EC2 : DescribeSecurityGroups
   - EC2 : DescribeSubnets
   - EC2 : DescribeVpcs
   - journaux : CreateLogGroup
   - journaux : CreateLogStream
   - journaux : PutLogEvents
   - responsable des secrets : GetSecretValue

   > **Note**
   >
   > Sans les autorisations IAM nécessaires, votre fonction ne sera pas en mesure de lire correctement les enregistrements des ressources Amazon MQ.

2. (Facultatif) Si vous avez créé un agent sans accès public, vous devez effectuer l'une des opérations suivantes pour permettre à Lambda de se connecter à votre agent :

   - Configurez une passerelle NAT par sous-réseau public. Pour plus d'informations, consultez Accès à Internet et aux services pour les fonctions connectées à un VPC dans le Guide du développeur AWS Lambda .
   - Créez une connexion entre votre Amazon Virtual Private Cloud (Amazon VPC) et Lambda à l'aide d'un point de terminaison VPC. Votre Amazon VPC doit également se connecter à AWS Security Token Service (AWS STS) et aux points de terminaison Secrets Manager. Pour plus d'informations, consultez Configuration de points de terminaison de VPC d'interface pour Lambda dans le Guide du développeur AWS Lambda .

3. Configurez votre agent en tant que source d'événement pour une fonction Lambda à l'aide de la AWS Management Console. Vous pouvez également utiliser la `create-event-source-mapping` AWS Command Line Interface commande.

4. Écrivez du code pour votre fonction Lambda pour traiter les messages de votre consommateur à partir de votre agent. La charge utile Lambda récupérée par votre mappage de source d'événement dépend du type de moteur de l'agent. Voici un exemple de charge utile Lambda pour une file d'attente Amazon MQ for RabbitMQ.

> **Note**
>
> Dans l'exemple, `test` est le nom de la file d'attente et `/` est le nom de l'hôte virtuel par défaut. Lors de la réception de messages, la source d'événement répertorie les messages sous `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=="
      }
    ]
  }
}
```

[Pour plus d'informations sur la connexion d'Amazon MQ à Lambda, sur les options prises en charge par Lambda pour une source d'événements Amazon MQ et sur les erreurs de mappage des sources d'événements, consultez la section Utilisation de Lambda avec Amazon MQ dans le manuel du développeur.AWS Lambda]

Utilisation de l'authentification et de l'autorisation OAuth 2.0 pour Amazon MQ pour RabbitMQ

Ce didacticiel explique comment configurer l'[authentification OAuth 2.0] pour votre Amazon MQ pour les courtiers RabbitMQ utilisant Amazon Cognito comme fournisseur 2.0. OAuth

> **Note**
>
> Amazon Cognito n'est pas disponible en Chine (Pékin) et en Chine (Ningxia).

> **Important**
>
> Ce didacticiel est spécifique à Amazon Cognito, mais vous pouvez utiliser d'autres fournisseurs d'identité ()IdPs. Pour plus d'informations, consultez la section Exemples d'authentification OAuth 2.0.

Sur cette page

Conditions préalables à la configuration de l'authentification OAuth 2.0

Vous pouvez définir les ressources Amazon Cognito requises dans ce didacticiel en déployant le plugin AWS CDK stack Amazon Cognito pour OAuth RabbitMQ 2. Si vous configurez Amazon Cognito manuellement, assurez-vous de remplir les conditions préalables suivantes avant de configurer la OAuth version 2.0 sur votre Amazon MQ pour les courtiers RabbitMQ :

Conditions préalables à la configuration d'Amazon Cognito

- Configurez un point de terminaison Amazon Cognito en créant un groupe d'utilisateurs. Pour ce faire, consultez le blog intitulé Comment utiliser la OAuth version 2.0 dans Amazon Cognito : découvrez les différentes subventions OAuth 2.0.
- Créez un serveur de ressources appelé `rabbitmq` dans le groupe d'utilisateurs avec les étendues suivantes définies :`read:all`, `write:allconfigure:all`, et`tag:administrator`. Ces étendues seront associées aux autorisations RabbitMQ.

  Pour plus d'informations sur la création d'un serveur de ressources, consultez la section Définition d'un serveur de ressources pour votre groupe d'utilisateurs (AWS Management Console) dans le manuel Amazon Cognito Developer Guide.
- Créez les clients d'application suivants :
  - Client d'application pour le groupe d'utilisateurs de ce type`Machine-to-Machine application`. Il s'agit d'un client confidentiel avec un secret client qui sera utilisé pour les clients RabbitMQ AMQP. Pour plus d'informations sur les clients d'applications et sur la création

d'un client d'application, consultez les sections Types de clients d'applications et Création d'un client d'application.

- Client d'application pour le groupe d'utilisateurs de ce type`Single-page application`. Il s'agit d'un client public qui sera utilisé pour connecter les utilisateurs à la console de gestion RabbitMQ. Vous devez mettre à jour ce client d'application pour inclure le point de terminaison du broker Amazon MQ pour RabbitMQ que vous allez créer dans la procédure suivante en tant qu'URL de rappel autorisée. Pour plus d'informations, consultez Configuration de la connexion gérée avec la console Amazon Cognito.

Prérequis pour configurer Amazon MQ

- Une installation Docker fonctionnelle pour exécuter un script bash qui vérifie si la configuration OAuth 2.0 est réussie ou non.
- AWS CLI version >= `2.28.23` pour rendre facultatif l'ajout d'un nom d'utilisateur et d'un mot de passe lors de la création du broker.

Configuration de l'authentification OAuth 2.0 avec Amazon Cognito à l'aide de AWS CLI

La procédure suivante explique comment configurer l'authentification OAuth 2.0 pour votre Amazon MQ pour les courtiers RabbitMQ utilisant Amazon Cognito comme IdP. Cette procédure permet AWS CLI de créer et de configurer les ressources nécessaires.

Dans la procédure suivante, assurez-vous de remplacer les valeurs d'espace réservé, telles que ConfigurationID et Revision*<2>*, *<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>* par leurs valeurs réelles.

1. Créez une nouvelle configuration à l'aide de la AWS CLI commande create-configuration comme indiqué dans l'exemple suivant.

```
aws mq create-configuration \
  --name "rabbitmq-oauth2-config" \
  --engine-type "RABBITMQ" \
  --engine-version "3.13"
```

   Cette commande renvoie une réponse similaire à l'exemple suivant.

```
{
    "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. Créez un fichier de configuration appelé **`rabbitmq.conf`** pour utiliser la OAuth version 2.0 comme méthode d'authentification et d'autorisation, comme indiqué dans l'exemple suivant.

```
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
```

Cette configuration utilise des alias d'étendue pour mapper les étendues définies dans Amazon Cognito à des étendues compatibles avec RabbitMQ.

3. Mettez à jour la configuration à l'aide de la AWS CLI commande update-configuration, comme indiqué dans l'exemple suivant. Dans cette commande, ajoutez l'ID de configuration que vous avez reçu en réponse à l'étape 1 de cette procédure. Par exemple, **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)"
```

Cette commande renvoie une réponse similaire à l'exemple suivant.

```
{
    "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. Créez un broker avec la configuration OAuth 2.0 que vous avez créée à l'étape 2 de cette procédure. Pour ce faire, utilisez la AWS CLI commande create-broker comme indiqué dans l'exemple suivant. Dans cette commande, indiquez l'ID de configuration et le numéro de révision que vous avez obtenus dans les réponses des étapes 1 et 2 respectivement. Par exemple : **c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca** et **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>}' \
```

Cette commande renvoie une réponse similaire à l'exemple suivant.

```
{
    "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. Vérifiez que le statut du courtier passe de à`RUNNING`, `CREATION_IN_PROGRESS` à l'aide de la AWS CLI commande describe-broker, comme indiqué dans l'exemple suivant. Dans cette commande, indiquez l'ID du courtier que vous avez obtenu dans le résultat de l'étape précédente. Par exemple,**`b-2a1b5133-a10c-49d2-879b-8c176c34cf73`**.

```
aws mq describe-broker \
 --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Cette commande renvoie une réponse similaire à l'exemple suivant. La réponse suivante est une version abrégée de la sortie complète renvoyée par la `describe-broker` commande. Cette réponse indique le statut du courtier et la stratégie d'authentification utilisée pour sécuriser le courtier. Dans ce cas, la stratégie `config_managed` d'authentification indique que le courtier utilise OAuth 2 méthodes d'authentification.

```
{
    "AuthenticationStrategy": "config_managed",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

Pour vous connecter à la console de gestion RabbitMQ en utilisant OAuth2, le point de terminaison du courtier doit être ajouté en tant qu'URL de rappel valide dans le client

d'application Amazon Cognito correspondant. Pour plus d'informations, reportez-vous à l'étape 5 de la configuration de notre exemple de stack Amazon Cognito CDK.

6. Vérifiez l'authentification et l'autorisation OAuth 2.0 à l'aide du `perf-test.sh` script suivant.

   Utilisez ce script bash pour tester la connectivité à votre courtier Amazon MQ for RabbitMQ. Ce script obtient un jeton auprès d'Amazon Cognito et vérifie si la connexion a été correctement configurée. S'il est correctement configuré, vous verrez votre courtier publier et consommer des messages.

   Si vous recevez un `ACCESS_REFUSED` message d'erreur, vous pouvez résoudre les problèmes liés à vos paramètres de configuration en utilisant les CloudWatch journaux de votre courtier. Vous trouverez le lien vers le groupe de CloudWatch journaux de votre courtier dans la console Amazon MQ.

   Dans ce script, vous devez fournir les valeurs suivantes :

   - `CLIENT_ID`et `CLIENT_SECRET` : vous pouvez trouver ces valeurs sur la page des clients de l'application de la console Amazon Cognito.
   - Domaine Cognito : vous pouvez le trouver sur la console Amazon Cognito. Sous Branding, sélectionnez Domain. Sur la page Domaine, vous pouvez trouver cette valeur dans la section Serveurs de ressources.
   - Point de terminaison du courtier Amazon MQ : vous pouvez trouver cette valeur sous Connexions sur la page de détails du courtier de la 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
```

Configuration de la OAuth version 2.0 et authentification simple avec Amazon Cognito

Lorsque vous créez un courtier avec une authentification OAuth 2.0, vous pouvez spécifier l'une des méthodes d'authentification suivantes :

- OAuth 2.0 uniquement : pour utiliser cette méthode, ne fournissez pas de nom d'utilisateur ni de mot de passe lors de la création du courtier. La procédure précédente montre comment utiliser uniquement la méthode d'authentification OAuth 2.0.
- Authentification OAuth 2.0 et authentification simple : pour utiliser cette méthode, fournissez un nom d'utilisateur et un mot de passe lors de la création du courtier. Ajoutez également `auth_backends.2 = internal` à la configuration de votre courtier, comme indiqué dans la procédure suivante.

Dans la procédure suivante, veillez à remplacer les valeurs d'espace réservé, telles que *<ConfigurationId>* et*<Revision>*, par leurs valeurs réelles.

1. Pour utiliser les deux méthodes d'authentification, créez la configuration de votre courtier, comme indiqué dans l'exemple suivant.

```
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:*/*
```

   Cette configuration utilise des alias d'étendue pour mapper les étendues définies dans Amazon Cognito à des étendues compatibles avec RabbitMQ.

2. Créez un courtier qui utilise les deux méthodes d'authentification, comme illustré dans l'exemple suivant.

```
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. Vérifiez que l'état du courtier et que la configuration de la méthode d'authentification a été correctement configurée, comme décrit aux étapes 5 et 6 de la Configuration de l'authentification OAuth 2.0 avec Amazon Cognito procédure.

Utilisation de l'authentification et de l'autorisation IAM pour Amazon MQ pour RabbitMQ

La procédure suivante explique comment activer l'authentification et l'autorisation AWS IAM pour un courtier Amazon MQ pour RabbitMQ. Après avoir activé IAM, les utilisateurs peuvent s'authentifier à l'aide des informations d'identification AWS IAM pour accéder à l'API de gestion RabbitMQ et se connecter via AMQP. Pour en savoir plus sur le fonctionnement de l'authentification IAM avec Amazon MQ pour RabbitMQ, consultez. the section called “Authentification et autorisation IAM”

Conditions préalables

- AWS informations d'identification d'administrateur pour le AWS compte propriétaire du courtier Amazon MQ pour RabbitMQ
- Un environnement shell configuré avec ces informations d'identification d'administrateur (à l'aide de profils AWS CLI ou de variables d'environnement)
- AWS CLI installée et configurée
- `jq`processeur JSON en ligne de commande installé
- `curl`outil de ligne de commande installé

Configuration de l'authentification et de l'autorisation IAM à l'aide de AWS CLI

1. Définir les variables d'environnement

   Définissez les variables d'environnement requises pour votre courtier :

   ```
   export AWS_DEFAULT_REGION=<region>
   export BROKER_ID=<broker-id>
   ```

2. Activer les jetons JWT sortants

   Activez la fédération d'identité Web sortante pour votre AWS compte :

```
ISSUER_IDENTIFIER=$(aws iam enable-outbound-web-identity-federation --query
 'IssuerIdentifier' --output text)
echo $ISSUER_IDENTIFIER
```

La sortie affiche une URL d'identifiant d'émetteur unique pour votre compte au format`https://<id>.tokens.sts.global.api.aws`.

3. Création du document de politique IAM

   Créez un document de politique qui accorde des autorisations pour obtenir des jetons d'identité Web :

```
cat > policy.json << 'EOF'
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "sts:GetWebIdentityToken",
                "sts:TagGetWebIdentityToken"
            ],
            "Resource": "*"
        }
    ]
}
EOF
```

4. Créez la politique de confiance

   Récupérez l'identité de votre appelant et créez un document de politique de confiance :

```
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. Création du rôle IAM

   Créez le rôle IAM et associez la politique :

```
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. Configurer les paramètres de RabbitMQ OAuth2

   Créez un fichier de configuration RabbitMQ avec les paramètres OAuth2 d'authentification et d'autorisation :

```
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. Mettre à jour la configuration du broker

   Appliquez la nouvelle configuration à votre courtier :

```
# 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
```

   Attendez que le statut du courtier revienne à nouveau `RUNNING` avant de passer à l'étape suivante.

8. Obtenir un jeton JWT

   Assumez le rôle IAM et obtenez un jeton d'identité 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. Accédez à l'API de gestion RabbitMQ

   Utilisez le jeton JWT pour accéder à l'API de gestion RabbitMQ :

```
BROKER_URL=<broker-id>.mq.<region>.on.aws

curl -u ":$TOKEN" \
    -X GET https://${BROKER_URL}/api/overview \
    -H "Content-Type: application/json"
```

   Une réponse réussie confirme que l'authentification IAM fonctionne correctement. La réponse contient des informations générales sur les courtiers au format JSON.

10. Connectez-vous via AMQP à l'aide du jeton JWT

    Testez la connectivité AMQP à l'aide du jeton JWT avec l'outil 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
```

Si vous recevez un `ACCESS_REFUSED` message d'erreur, vous pouvez résoudre les problèmes liés à vos paramètres de configuration en utilisant les CloudWatch journaux de votre courtier. Vous trouverez le lien vers le groupe de CloudWatch journaux Logs de votre courtier dans la console Amazon MQ.

Utilisation de l'authentification et de l'autorisation LDAP pour Amazon MQ pour RabbitMQ

Ce didacticiel explique comment configurer l'authentification et l'autorisation LDAP pour votre Amazon MQ pour les courtiers RabbitMQ à l'aide de. AWS Managed Microsoft AD

Sur cette page

Conditions préalables pour configurer l'authentification et l'autorisation LDAP

Vous pouvez configurer les AWS ressources requises dans ce didacticiel en déployant la pile AWS CDK pour Amazon MQ pour l'intégration LDAP de RabbitMQ avec. AWS Managed Microsoft AD

Cette pile CDK crée automatiquement toutes les AWS ressources nécessaires AWS Managed Microsoft AD, notamment les utilisateurs et groupes LDAP, Network Load Balancer, les certificats et les rôles IAM. Consultez le package README pour une liste complète des ressources créées par la pile.

Si vous configurez les ressources manuellement au lieu d'utiliser la pile CDK, assurez-vous de disposer de l'infrastructure équivalente avant de configurer LDAP sur votre Amazon MQ pour les courtiers RabbitMQ.

Prérequis pour configurer Amazon MQ

AWS Version CLI >= 2.28.23 pour rendre l'ajout d'un nom d'utilisateur et d'un mot de passe facultatif lors de la création du broker.

Configuration de LDAP dans RabbitMQ à l'aide de la CLI AWS

Cette procédure utilise la AWS CLI pour créer et configurer les ressources nécessaires. Dans la procédure suivante, assurez-vous de remplacer les valeurs d'espace réservé, telles que ConfigurationID et Revision<2>, <c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca> par leurs valeurs réelles.

1. Créez une nouvelle configuration à l'aide de la commande `create-configuration` AWS CLI comme indiqué dans l'exemple suivant.

```
aws mq create-configuration \
  --name "rabbitmq-ldap-config" \
  --engine-type "RABBITMQ" \
  --engine-version "3.13"
```

   Cette commande renvoie une réponse similaire à l'exemple suivant.

```
{
"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. Créez un fichier de configuration appelé `rabbitmq.conf` pour utiliser LDAP comme méthode d'authentification et d'autorisation, comme indiqué dans l'exemple suivant. Remplacez toutes les valeurs d'espace réservé du modèle (marquées d'un`${RabbitMqLdapTestStack.*}`) par des valeurs réelles provenant de vos sorties de pile AWS CDK prérequises déployées ou d'une infrastructure équivalente.

```
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. Mettez à jour la configuration à l'aide de la commande `update-configuration` AWS CLI comme indiqué dans l'exemple suivant. Dans cette commande, ajoutez l'ID de configuration que vous avez reçu en réponse à l'étape 1 de cette procédure. Par exemple, `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)"
```

Cette commande renvoie une réponse similaire à l'exemple suivant.

```
{
    "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. Créez un courtier avec la configuration LDAP que vous avez créée à l'étape 2 de cette procédure. Pour ce faire, utilisez la commande `create-broker` AWS CLI comme indiqué dans l'exemple suivant. Dans cette commande, indiquez l'ID de configuration et le numéro de révision que vous avez obtenus dans les réponses des étapes 1 et 2 respectivement. Par exemple : `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` et 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>}'
```

Cette commande renvoie une réponse similaire à l'exemple suivant.

```
{
    "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"
```

```
}
```

> **Restriction de dénomination des courtiers**
>
> Le rôle IAM créé par la pile CDK requise limite d'emblée les noms des courtiers. `rabbitmq-ldap-test` Assurez-vous que le nom de votre courtier suit ce modèle, sinon le rôle IAM ne sera pas autorisé à assumer le rôle de résolution des ARN.

5. Vérifiez que le statut du broker passe de à`RUNNING`, `CREATION_IN_PROGRESS` à l'aide de la commande `describe-broker` AWS CLI, comme indiqué dans l'exemple suivant. Dans cette commande, indiquez l'ID du courtier que vous avez obtenu dans le résultat de l'étape précédente. Par exemple,`b-2a1b5133-a10c-49d2-879b-8c176c34cf73`.

```
aws mq describe-broker \
 --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Cette commande renvoie une réponse similaire à l'exemple suivant. La réponse suivante est une version abrégée de la sortie complète renvoyée par la `describe-broker` commande. Cette réponse indique le statut du courtier et la stratégie d'authentification utilisée pour sécuriser le courtier. Dans ce cas, la stratégie `config_managed` d'authentification indique que le courtier utilise la méthode d'authentification LDAP.

```
{
"AuthenticationStrategy": "config_managed",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Validez l'accès à RabbitMQ en utilisant l'un des utilisateurs de test créés par la pile CDK requise

```
# 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"}'
```

Utilisation de l'authentification et de l'autorisation HTTP pour Amazon MQ pour RabbitMQ

Ce didacticiel explique comment configurer l'authentification et l'autorisation HTTP pour votre Amazon MQ pour les courtiers RabbitMQ à l'aide d'un serveur HTTP externe.

> **Note**
>
> Le plug-in d'authentification HTTP est uniquement disponible pour Amazon MQ pour RabbitMQ version 4 et supérieure.

Sur cette page

Conditions préalables à la configuration de l'authentification et de l'autorisation HTTP

Vous pouvez configurer les AWS ressources requises dans ce didacticiel en déployant la pile AWS CDK pour Amazon MQ pour l'intégration de l'authentification HTTP RabbitMQ.

Cette pile CDK crée automatiquement toutes les AWS ressources nécessaires, y compris le serveur d'authentification HTTP, les certificats et les rôles IAM. Consultez le package README pour une liste complète des ressources créées par la pile.

Si vous configurez les ressources manuellement au lieu d'utiliser la pile CDK, assurez-vous de disposer de l'infrastructure équivalente avant de configurer l'authentification HTTP sur votre Amazon MQ pour les courtiers RabbitMQ.

Prérequis pour configurer Amazon MQ

AWS Version CLI >= 2.28.23 pour rendre l'ajout d'un nom d'utilisateur et d'un mot de passe facultatif lors de la création du broker.

Configuration de l'authentification HTTP dans RabbitMQ à l'aide de la CLI AWS

Cette procédure utilise la AWS CLI pour créer et configurer les ressources nécessaires. Dans la procédure suivante, assurez-vous de remplacer les valeurs de l'espace réservé par leurs valeurs réelles.

1. Créez une nouvelle configuration à l'aide de la commande `create-configuration` AWS CLI, comme indiqué dans l'exemple suivant.

```
aws mq create-configuration \
  --name "rabbitmq-http-config" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2"
```

   Cette commande renvoie une réponse similaire à l'exemple suivant.

```
{
    "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. Créez un fichier de configuration appelé `rabbitmq.conf` pour utiliser le protocole HTTP comme méthode d'authentification et d'autorisation, comme indiqué dans l'exemple suivant. Remplacez toutes les valeurs d'espace réservé du modèle (marquées d'un`${...}`) par des valeurs réelles provenant de vos sorties de pile AWS CDK prérequises déployées ou d'une infrastructure équivalente.

```
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. Mettez à jour la configuration à l'aide de la commande `update-configuration` AWS CLI. Utilisez l'ID de configuration indiqué à l'étape 3.

```
aws mq update-configuration \
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Cette commande renvoie une réponse similaire à l'exemple suivant.

```
{
    "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. Créez un broker avec la configuration HTTP. Utilisez l'ID de configuration et le numéro de révision indiqués lors des étapes précédentes.

```
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>}'
```

Cette commande renvoie une réponse similaire à l'exemple suivant.

```
{
    "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. Vérifiez que le statut du broker passe de àRUNNING, `CREATION_IN_PROGRESS` à l'aide de la commande `describe-broker` AWS CLI.

```
aws mq describe-broker \
  --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Cette commande renvoie une réponse similaire à l'exemple suivant. La stratégie `config_managed` d'authentification indique que le courtier utilise la méthode d'authentification HTTP.

```
{
    "AuthenticationStrategy": "config_managed",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Validez l'accès à RabbitMQ en utilisant l'un des utilisateurs de test créés par la pile CDK requise

```
# 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 '{}'
```

Utilisation de l'authentification par certificat SSL pour Amazon MQ pour RabbitMQ

Ce didacticiel explique comment configurer l'authentification par certificat SSL pour votre Amazon MQ pour les courtiers RabbitMQ à l'aide d'une autorité de certification privée.

> **Note**
>
> Le plug-in d'authentification par certificat SSL est uniquement disponible pour Amazon MQ pour RabbitMQ version 4 et supérieure.

Sur cette page

Conditions préalables à la configuration de l'authentification par certificat SSL

L'authentification par certificat SSL utilise le protocole TLS mutuel (mTLS) pour authentifier les clients à l'aide de certificats X.509. Vous pouvez configurer les AWS ressources requises dans ce didacticiel en déployant la pile AWS CDK pour Amazon MQ pour l'intégration mTLS de RabbitMQ.

Cette pile CDK crée automatiquement toutes les AWS ressources nécessaires, y compris l'autorité de certification, les certificats clients et les rôles IAM. Consultez le package README pour une liste complète des ressources créées par la pile.

> **Note**
>
> Avant de déployer la pile CDK, définissez la variable d'`RABBITMQ_TEST_USER_NAME`environnement. Cette valeur sera utilisée comme nom commun (CN) dans le certificat client et doit correspondre au nom d'utilisateur que vous utilisez dans les étapes du didacticiel. Par exemple : `export RABBITMQ_TEST_USER_NAME="myuser"`

Si vous configurez les ressources manuellement au lieu d'utiliser la pile CDK, assurez-vous de disposer de l'infrastructure équivalente avant de configurer l'authentification par certificat SSL sur votre Amazon MQ pour les courtiers RabbitMQ.

Prérequis pour configurer Amazon MQ

AWS Version CLI >= 2.28.23 pour rendre l'ajout d'un nom d'utilisateur et d'un mot de passe facultatif lors de la création du broker.

Configuration de l'authentification par certificat SSL dans RabbitMQ à l'aide de la CLI AWS

Cette procédure utilise la AWS CLI pour créer et configurer les ressources nécessaires. Dans la procédure suivante, assurez-vous de remplacer les valeurs d'espace réservé, telles que ConfigurationID et Revision<2>, `<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>` par leurs valeurs réelles.

1. Créez une nouvelle configuration à l'aide de la commande `create-configuration` AWS CLI, comme indiqué dans l'exemple suivant.

```
aws mq create-configuration \
  --name "rabbitmq-ssl-config" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2"
```

   Cette commande renvoie une réponse similaire à l'exemple suivant.

```
{
    "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. Créez un fichier de configuration appelé `rabbitmq.conf` pour utiliser l'authentification par certificat SSL, comme indiqué dans l'exemple suivant. Remplacez toutes les valeurs d'espace réservé du modèle (marquées d'un`${...}`) par des valeurs réelles provenant de vos sorties de pile AWS CDK prérequises déployées ou d'une infrastructure équivalente.

```
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. Mettez à jour la configuration à l'aide de la commande `update-configuration` AWS CLI comme indiqué dans l'exemple suivant. Dans cette commande, ajoutez l'ID de

configuration que vous avez reçu en réponse à l'étape 1 de cette procédure. Par exemple, `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)"
```

Cette commande renvoie une réponse similaire à l'exemple suivant.

```
{
    "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. Créez un courtier avec la configuration d'authentification par certificat SSL que vous avez créée à l'étape 2 de cette procédure. Pour ce faire, utilisez la commande `create-broker` AWS CLI comme indiqué dans l'exemple suivant. Dans cette commande, indiquez l'ID de configuration et le numéro de révision que vous avez obtenus dans les réponses des étapes 1 et 2 respectivement. Par exemple : `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` et 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"}]'
```

Cette commande renvoie une réponse similaire à l'exemple suivant.

```
{
    "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. Vérifiez que le statut du broker passe de àRUNNING, CREATION_IN_PROGRESS à l'aide de la commande `describe-broker` AWS CLI, comme indiqué dans l'exemple suivant. Dans cette commande, indiquez l'ID du courtier que vous avez obtenu dans le résultat de l'étape précédente. Par exemple, `b-2a1b5133-a10c-49d2-879b-8c176c34cf73`.

```
aws mq describe-broker \
  --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Cette commande renvoie une réponse similaire à l'exemple suivant. La réponse suivante est une version abrégée de la sortie complète renvoyée par la `describe-broker` commande. Cette réponse indique le statut du courtier et la stratégie d'authentification utilisée pour sécuriser le courtier. Dans ce cas, la stratégie `config_managed` d'authentification indique que le courtier utilise la méthode d'authentification par certificat SSL.

```
{
    "AuthenticationStrategy": "config_managed",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Vérifiez l'authentification du certificat SSL à l'aide du `ssl.sh` script suivant.

Utilisez ce script bash pour tester la connectivité à votre courtier Amazon MQ for RabbitMQ. Ce script utilise votre certificat client pour l'authentification et vérifie si la connexion a été correctement configurée. S'il est correctement configuré, vous verrez votre courtier publier et consommer des messages.

Si vous recevez un `ACCESS_REFUSED` message d'erreur, vous pouvez résoudre les problèmes liés à vos paramètres de configuration en utilisant les CloudWatch journaux de votre courtier. Vous trouverez le lien vers le groupe de CloudWatch journaux de votre courtier dans la console Amazon MQ.

Dans ce script, vous devez fournir les valeurs suivantes :

- `USERNAME`: le nom commun (CN) de votre certificat client.
- `CLIENT_KEYSTORE`: chemin d'accès au fichier keystore de votre client (PKCS12 format). Si vous avez utilisé la pile CDK requise, le chemin par défaut est`$(pwd)/certs/client-keystore.p12`.
- `KEYSTORE_PASSWORD`: mot de passe pour le keystore de votre client. Si vous avez utilisé la pile CDK requise, le mot de passe par défaut est`changeit`.
- `BROKER_DNS`: vous pouvez trouver cette valeur sous Connexions sur la page des informations du courtier de la 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
```

Utilisation de MTL pour l'AMQP et les points de terminaison de gestion

Ce didacticiel explique comment configurer le protocole TLS mutuel (MTL) pour les connexions client AMQP et l'interface de gestion RabbitMQ à l'aide d'une autorité de certification privée.

> **Note**
>
> L'utilisation d'autorités de certification privées pour les MTL n'est disponible que pour Amazon MQ pour RabbitMQ version 4 et supérieure.

Sur cette page

Conditions préalables à la configuration des mTLs

Vous pouvez configurer les AWS ressources requises dans ce didacticiel en déployant la pile AWS CDK pour Amazon MQ pour l'intégration de mTLS avec RabbitMQ.

Cette pile CDK crée automatiquement toutes les AWS ressources nécessaires, y compris l'autorité de certification, les certificats clients et les rôles IAM. Consultez le package README pour une liste complète des ressources créées par la pile.

Si vous configurez les ressources manuellement au lieu d'utiliser la pile CDK, assurez-vous de disposer de l'infrastructure équivalente avant de configurer les MTL sur votre Amazon MQ pour les courtiers RabbitMQ.

Prérequis pour configurer Amazon MQ

AWS Version CLI >= 2.28.23 pour rendre l'ajout d'un nom d'utilisateur et d'un mot de passe facultatif lors de la création du broker.

Configuration de MTL dans RabbitMQ à l'aide de la CLI AWS

Cette procédure utilise la AWS CLI pour créer et configurer les ressources nécessaires. Dans la procédure suivante, assurez-vous de remplacer les valeurs d'espace réservé, telles que ConfigurationID et Revision<2>, <c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca> par leurs valeurs réelles.

1. Créez une nouvelle configuration à l'aide de la commande `create-configuration` AWS CLI comme indiqué dans l'exemple suivant.

```
aws mq create-configuration \
  --name "rabbitmq-mtls-config" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2"
```

   Cette commande renvoie une réponse similaire à l'exemple suivant.

```
{
    "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. Créez un fichier de configuration appelé `rabbitmq.conf` pour configurer les MTL pour l'AMQP et les points de terminaison de gestion, comme indiqué dans l'exemple suivant. Remplacez toutes les valeurs d'espace réservé du modèle (marquées d'un`${...}`) par des valeurs réelles provenant de vos sorties de pile AWS CDK prérequises déployées ou d'une infrastructure équivalente.

```
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. Mettez à jour la configuration à l'aide de la commande `update-configuration` AWS CLI comme indiqué dans l'exemple suivant. Dans cette commande, ajoutez l'ID de configuration que vous avez reçu en réponse à l'étape 1 de cette procédure. Par exemple, `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)"
```

Cette commande renvoie une réponse similaire à l'exemple suivant.

```
{
    "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. Créez un broker avec la configuration mTLS que vous avez créée à l'étape 2 de cette procédure. Pour ce faire, utilisez la commande `create-broker` AWS CLI comme indiqué dans l'exemple suivant. Dans cette commande, indiquez l'ID de configuration et le numéro de révision que vous avez obtenus dans les réponses des étapes 1 et 2 respectivement. Par exemple : `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` et 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"}]'
```

Cette commande renvoie une réponse similaire à l'exemple suivant.

```
{
    "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. Vérifiez que le statut du broker passe de àRUNNING, CREATION_IN_PROGRESS à l'aide de la commande `describe-broker` AWS CLI, comme indiqué dans l'exemple suivant. Dans cette commande, indiquez l'ID du courtier que vous avez obtenu dans le résultat de l'étape précédente. Par exemple, `b-2a1b5133-a10c-49d2-879b-8c176c34cf73`.

```
aws mq describe-broker \
  --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Cette commande renvoie une réponse similaire à l'exemple suivant. La réponse suivante est une version abrégée de la sortie complète renvoyée par la `describe-broker` commande.

```
{
    "AuthenticationStrategy": "simple",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Vérifiez l'authentification mTLS à l'aide du `mtls.sh` script suivant.

Utilisez ce script bash pour tester la connectivité à votre courtier Amazon MQ for RabbitMQ. Ce script utilise votre certificat client pour authentifier et vérifier si la connexion a été correctement configurée. S'il est correctement configuré, vous verrez votre courtier publier et consommer des messages.

Si vous recevez un ACCESS_REFUSED message d'erreur, vous pouvez résoudre les problèmes liés à vos paramètres de configuration en utilisant les CloudWatch journaux de votre courtier.

Vous trouverez le lien vers le groupe de CloudWatch journaux de votre courtier dans la console Amazon MQ.

Dans ce script, vous devez fournir les valeurs suivantes :

- `USERNAME`et `PASSWORD` : les informations d'identification utilisateur RabbitMQ que vous avez créées avec le courtier.
- `CLIENT_KEYSTORE`: chemin d'accès au fichier keystore de votre client (PKCS12 format). Si vous avez utilisé la pile CDK requise, le chemin par défaut est`$(pwd)/certs/client-keystore.p12`.
- `KEYSTORE_PASSWORD`: mot de passe pour le keystore de votre client. Si vous avez utilisé la pile CDK requise, le mot de passe par défaut est`changeit`.
- `BROKER_DNS`: vous pouvez trouver cette valeur sous Connexions sur la page des informations du courtier de la 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
```

Connexion de votre application JMS

Ce didacticiel explique comment connecter votre application JMS au courtier Amazon MQ pour RabbitMQ à l'aide du client JMS RabbitMQ. Vous apprendrez à créer un producteur pour envoyer des messages et un consommateur pour recevoir des messages depuis les files d'attente de RabbitMQ.

Avant de commencer, ajoutez la dépendance RabbitMQ JMS appropriée à votre projet Maven :

Pour JMS 1.1 et 2.0 :

```
<dependencies>

  <dependency>
    <groupId>com.rabbitmq.jms</groupId>
    <artifactId>rabbitmq-jms</artifactId>
    <version>2.12.0</version>
  </dependency>

</dependencies>
```

Pour JMS 3.1 :

```
<dependencies>

  <dependency>
    <groupId>com.rabbitmq.jms</groupId>
    <artifactId>rabbitmq-jms</artifactId>
    <version>3.5.0</version>
  </dependency>

</dependencies>
```

Création d'un producteur

L'exemple de code suivant montre comment écrire dans une file d'attente RabbitMQ à l'aide de 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);
```

Créez un consommateur

L'exemple de code suivant montre comment lire à partir d'une file d'attente RabbitMQ à l'aide de 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();
```

Sécurité dans Amazon MQ

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à Amazon MQ, consultez AWS Services concernés par programme de conformitéAWS Services couverts par programme .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre entreprise, et la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de Amazon MQ. Les rubriques suivantes vous montrent comment configurer Amazon MQ pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos ressources Amazon MQ.

Rubriques

Protection des données dans Amazon MQ

Le modèle de responsabilité AWS partagée s'applique à la protection des données dans Amazon MQ. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée d'AWS et RGPD (Règlement général sur la protection des données) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou Gestion des identités et des accès AWS (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- SSL/TLS À utiliser pour communiquer avec AWS les ressources. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section Utilisation des CloudTrail sentiers dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez Norme FIPS (Federal Information Processing Standard) 140-3.

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec Amazon MQ ou une autre entreprise à

Services AWS l'aide de la console, de l'API ou. AWS CLI AWS SDKs Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Pour les agents Amazon MQ for ActiveMQ et Amazon MQ for RabbitMQ, n'utilisez pas de données d'identification personnelle (PII) ou d'autres données confidentielles ou sensibles pour les noms d'agents ou les noms d'utilisateurs lorsque vous créez des ressources via la console web de l'agent ou l'API Amazon MQ. Les noms des courtiers et les noms d'utilisateur sont accessibles à d'autres AWS services, notamment aux CloudWatch journaux. Les noms d'utilisateur des agents ne sont pas destinées à être utilisées pour des données privées ou sensibles.

> **Important**
>
> TLS 1.3 n'est pas disponible pour les courtiers RabbitMQ.

Chiffrement

Les données utilisateur stockées dans Amazon MQ sont chiffrées au repos. Le chiffrement au repos Amazon MQ offre une sécurité renforcée en chiffrant vos données au repos à l'aide de clés de chiffrement stockées dans AWS Key Management Service (KMS). Ce service réduit la lourdeur opérationnelle et la complexité induites par la protection des données sensibles. Le chiffrement au repos vous permet de créer des applications sensibles en matière de sécurité qui sont conformes aux exigences réglementaires et de chiffrement.

Toutes les connexions entre les agents Amazon MQ utilisent le protocole TLS (Transport layer Security) pour assurer le chiffrement en transit.

Amazon MQ chiffre les messages au repos et en transit à l'aide de clés de chiffrement qu'il gère et stocke en toute sécurité. Pour plus d'informations, consultez le Guide du développeur AWS Encryption SDK.

Chiffrement au repos

Amazon MQ s'intègre à AWS Key Management Service (KMS) pour offrir un chiffrement transparent côté serveur. Amazon MQ chiffre toujours vos données au repos.

Lorsque vous créez un courtier Amazon MQ pour ActiveMQ ou un courtier Amazon MQ pour RabbitMQ, vous pouvez spécifier celui que vous AWS KMS key souhaitez qu'Amazon MQ utilise pour chiffrer vos données au repos. Si vous ne spécifiez pas de clé KMS, Amazon MQ crée une clé KMS AWS propriétaire pour vous et l'utilise en votre nom. Amazon MQ prend en charge actuellement les clés KMS symétriques. Pour plus d'informations sur les clés KMS, consultez AWS KMS keys.

Lorsque vous créez un agent, vous pouvez configurer la clé de chiffrement utilisée par Amazon MQ en sélectionnant l'une des options suivantes.

- Clé KMS détenue par Amazon MQ (valeur par défaut) – La clé est détenue par Amazon MQ et ne figure pas dans votre compte.
- AWS clé KMS AWS gérée : la clé KMS gérée (`aws/mq`) est une clé KMS de votre compte créée, gérée et utilisée en votre nom par Amazon MQ.
- Sélection d'une clé KMS existante gérée par le client – Vous créez et gérez les clés KMS gérées par le client dans AWS Key Management Service (KMS).

⚠ Important

- La révocation d'un octroi ne peut pas être annulée. Supprimez le courtier pour révoquer les droits d'accès.
- Pour les courtiers Amazon MQ for ActiveMQ qui utilisent Amazon Elastic File System (EFS) pour stocker les données des messages, la révocation des autorisations d'utilisation des clés KMS de votre compte peut prendre plusieurs heures après avoir pris les mesures requises.
- Pour les agents Amazon MQ for RabbitMQ et Amazon MQ for ActiveMQ qui utilisent EBS pour stocker les données des messages, si vous désactivez, planifiez la suppression ou révoquez l'octroi qui autorise Amazon EBS à utiliser les clés KMS contenues dans votre compte, Amazon MQ ne peut pas conserver votre agent et celui-ci peut devenir dégradé.
- Si vous avez désactivé la clé ou planifié sa suppression, vous pouvez la réactiver ou annuler la suppression de la clé et conserver votre agent.
- Plusieurs heures peuvent être nécessaires pour désactiver une clé ou révoquer une autorisation après avoir pris les mesures requises.
- Pour chiffrer ou déchiffrer les CloudWatch journaux, vous ne pouvez pas configurer la clé de chiffrement utilisée par Amazon MQ. CloudWatch les journaux protègent les données au repos grâce au chiffrement, et les groupes de journaux sont chiffrés. Le

service de CloudWatch journalisation gère le chiffrement côté serveur par défaut. Pour plus d'informations sur le chiffrement des groupes de journaux, consultez le guide de l'utilisateur d'Amazon CloudWatch Logs.

Lors de la création d'un agent d'instance unique avec une clé KMS pour RabbitMQ, vous voyez deux événements `CreateGrant` connectés dans AWS CloudTrail. Le premier événement correspond à la création par Amazon MQ d'une autorisation pour la clé KMS. Le deuxième événement correspond à la création par EBS d'une autorisation qu'EBS pourra utiliser.

CreateGrant AWS CloudTrail entrée de journal : courtier à instance unique

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

Vous verrez un événement unique pour la création d'une autorisation 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"
}
```

Lors de la création d'un déploiement de cluster avec une clé KMS pour RabbitMQ, vous voyez cinq événements `CreateGrant` connectés dans AWS CloudTrail. Les deux premiers événements sont des créations d'autorisation pour Amazon MQ. Les trois événements suivants sont des autorisations créées par EBS qu'EBS pourra utiliser.

CreateGrant AWS CloudTrail entrée de journal : déploiement du 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

Vous verrez trois événements pour la création d'autorisations 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"
}
```

Pour plus d'informations sur les clés KMS, consultez AWS KMS keys dans le Guide du développeur AWS Key Management Service .

Chiffrement en transit

Amazon MQ for ActiveMQ : Amazon MQ for ActiveMQ nécessite un protocole TLS (Transport Layer Security) renforcé et chiffre les données en transit entre les agents de votre déploiement Amazon MQ. Toutes les données transmises entre les agents Amazon MQ sont chiffrées à l'aide du protocole TLS (Transport Layer Security) sécurisé. Cette règle s'applique à tous les protocoles disponibles.

Amazon MQ for RabbitMQ : Amazon MQ for RabbitMQ nécessite un chiffrement par protocole TLS (Transport Layer Security) sécurisé pour toutes les connexions client. Le trafic de réplication du cluster RabbitMQ transite uniquement par le VPC de votre courtier et tout le trafic réseau entre les centres de AWS données est crypté de manière transparente au niveau de la couche physique. Les agents en cluster Amazon MQ for RabbitMQ ne prennent actuellement pas en charge le chiffrement entre nœuds pour la réplication en cluster. Pour en savoir plus data-in-transit, consultez la section Chiffrement Data-at-Rest et -in-Transit.

Protocoles Amazon MQ for ActiveMQ

Vous pouvez accéder à vos agents ActiveMQ en utilisant les protocoles suivants avec TLS activé :

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

Suites de chiffrement TLS pour ActiveMQ prises en charge

ActiveMQ sur Amazon MQ prend en charge les suites de chiffrement suivantes :

- TLS_ECDHE_RSA_WITH_AES_256_GCM_ SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_ SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

- TLS_DHE_RSA_WITH_AES_256_GCM_ SHA384
- TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_256_GCM_ SHA384
- TLS_RSA_WITH_AES_256_CBC_ SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_GCM_ SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_ SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_GCM_ SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_128_GCM_ SHA256
- TLS_RSA_WITH_AES_128_CBC_ SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA

Protocoles Amazon MQ for RabbitMQ

Vous pouvez accéder à vos agents RabbitMQ en utilisant les protocoles suivants avec TLS activé :

- AMQP (0-9-1)

Suites de chiffrement TLS pour RabbitMQ prises en charge

RabbitMQ sur Amazon MQ prend en charge les suites de chiffrement suivantes :

- TLS_ECDHE_RSA_WITH_AES_256_GCM_ SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_ SHA256

Identity and Access Management pour Amazon MQ

Gestion des identités et des accès AWS (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Des administrateurs

IAM contrôlent les personnes qui peuvent être authentifiées(connectées) et autorisées (disposant d'autorisations) pour utiliser des ressources Amazon MQ. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

Public ciblé

La façon dont vous utilisez Gestion des identités et des accès AWS (IAM) varie en fonction de votre rôle :

- Utilisateur du service : demandez des autorisations à votre administrateur si vous ne pouvez pas accéder aux fonctionnalités (voir Résolution de problèmes pour identité et accès Amazon MQ)
- Administrateur du service : déterminez l’accès des utilisateurs et soumettez les demandes d’autorisation (voir Fonctionnement d'Amazon MQ avec IAM)
- Administrateur IAM : rédigez des politiques pour gérer l’accès (voir Exemples de politique basée sur l'identité d'Amazon MQ)

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en tant qu'identité fédérée à l'aide d'informations d'identification provenant d'une source d'identité telle que AWS IAM Identity Center (IAM Identity Center), d'une authentification unique ou d'informations d'identification. Google/Facebook Pour plus d'informations sur la connexion, consultez Connexion à votre Compte AWS dans le Guide de l'utilisateur Connexion à AWS .

Pour l'accès par programmation, AWS fournit un SDK et une CLI pour signer les demandes de manière cryptographique. Pour plus d'informations, consultez Signature AWS Version 4 pour les demandes d'API dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une seule identité de connexion appelée utilisateur Compte AWS root qui dispose d'un accès complet à toutes Services AWS les ressources. Il est vivement déconseillé d'utiliser l'utilisateur racine pour vos tâches quotidiennes. Pour les tâches qui requièrent des informations d'identification de l'utilisateur racine, consultez Tâches qui requièrent les informations d'identification de l'utilisateur racine dans le Guide de l'utilisateur IAM.

Utilisateurs et groupes

Un utilisateur IAM est une identité qui dispose d'autorisations spécifiques pour une seule personne ou application. Nous vous recommandons d'utiliser ces informations d'identification temporaires au lieu des utilisateurs IAM avec des informations d'identification à long terme. Pour plus d'informations, voir Exiger des utilisateurs humains qu'ils utilisent la fédération avec un fournisseur d'identité pour accéder à AWS l'aide d'informations d'identification temporaires dans le guide de l'utilisateur IAM.

Les groupes IAM spécifient une collection d'utilisateurs IAM et permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Pour plus d'informations, consultez Cas d'utilisation pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité dotée d'autorisations spécifiques qui fournit des informations d'identification temporaires. Vous pouvez assumer un rôle en passant d'un rôle d'utilisateur à un rôle IAM (console) ou en appelant une opération d' AWS API AWS CLI ou d'API. Pour plus d'informations, consultez Méthodes pour endosser un rôle dans le Guide de l'utilisateur IAM.

Les rôles IAM sont utiles pour l'accès des utilisateurs fédérés, les autorisations temporaires des utilisateurs IAM, les accès intercompte, les accès entre services et les applications exécutées sur

Amazon EC2. Pour plus d'informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Gestion de l'accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique définit les autorisations lorsqu'elles sont associées à une identité ou à une ressource. AWS évalue ces politiques lorsqu'un directeur fait une demande. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations les documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

À l'aide de politiques, les administrateurs précisent qui a accès à quoi en définissant quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Un administrateur IAM crée des politiques IAM et les ajoute aux rôles, que les utilisateurs peuvent ensuite assumer. Les politiques IAM définissent les autorisations quelle que soit la méthode que vous utilisez pour exécuter l'opération.

Politiques basées sur l'identité

Les stratégies basées sur l'identité sont des documents de stratégie d'autorisations JSON que vous attachez à une identité (utilisateur, groupe ou rôle). Ces politiques contrôlent les actions que peuvent exécuter ces identités, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être des politiques intégrées (intégrées directement dans une seule identité) ou des politiques gérées (politiques autonomes associées à plusieurs identités). Pour découvrir comment choisir entre des politiques gérées et en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Les exemples incluent les politiques de confiance de rôle IAM et les stratégies de compartiment Amazon S3. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Vous devez spécifier un principal dans une politique basée sur les ressources.

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Listes de contrôle d'accès (ACLs)

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et AWS WAF Amazon VPC sont des exemples de services compatibles. ACLs Pour en savoir plus ACLs, consultez la présentation de la liste de contrôle d'accès (ACL) dans le guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge des types de politiques supplémentaires qui peuvent définir les autorisations maximales accordées par les types de politiques les plus courants :

- Limites d’autorisations : une limite des autorisations définit le nombre maximum d’autorisations qu’une politique basée sur l’identité peut accorder à une entité IAM. Pour plus d’informations, consultez Limites d’autorisations pour des entités IAM dans le Guide de l’utilisateur IAM.
- Politiques de contrôle des services (SCPs) — Spécifiez les autorisations maximales pour une organisation ou une unité organisationnelle dans AWS Organizations. Pour plus d’informations, consultez Politiques de contrôle de service dans le Guide de l’utilisateur AWS Organizations .
- Politiques de contrôle des ressources (RCPs) : définissez le maximum d'autorisations disponibles pour les ressources de vos comptes. Pour plus d'informations, voir Politiques de contrôle des ressources (RCPs) dans le guide de AWS Organizations l'utilisateur.
- Politiques de session : politiques avancées que vous passez en tant que paramètre lorsque vous créez par programmation une session temporaire pour un rôle ou un utilisateur fédéré. Pour plus d’informations, consultez Politiques de session dans le Guide de l’utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s’appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

Fonctionnement d'Amazon MQ avec IAM

Avant d'utiliser IAM pour gérer l'accès à Amazon MQ, vous devez comprendre quelles sont les fonctionnalités IAM qui peuvent être utilisées dans cette situation. Pour obtenir une vue d'ensemble de la manière dont Amazon MQ et les autres AWS services fonctionnent avec IAM, consultez AWS la section Services That Work with IAM dans le guide de l'utilisateur d'IAM.

Amazon MQ utilise IAM pour les opérations d'API Amazon MQ afin de créer, mettre à jour, supprimer et répertorier des courtiers. Pour permettre aux courtiers de publier des messages et de s'y abonner, Amazon MQ pour ActiveMQ prend en charge l'authentification ActiveMQ native et le protocole LDAP, tandis qu'Amazon MQ pour RabbitMQ prend en charge l'authentification IAM et d'autres méthodes. Pour de plus amples informations, veuillez consulter the section called “Authentification et autorisation du courtier”.

Rubriques

- Politiques basées sur l'identité Amazon MQ
- Politiques basées sur des ressources Amazon MQ
- Autorisation basée sur des balises Amazon MQ
- Rôles IAM Amazon MQ

Politiques basées sur l'identité Amazon MQ

Avec les politiques IAM basées sur l’identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Amazon MQ est compatible avec des actions, des ressources et des clés de condition spécifiques. Pour en savoir plus sur tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l’utilisateur IAM.

Actions

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C’est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L’élément `Action` d’une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l’accès à une politique. Intégration d’actions dans une politique afin d’accorder l’autorisation d’exécuter les opérations associées.

Les actions de politique dans Amazon MQ utilisent le préfixe suivant avant l'action : `mq:`. Par exemple, pour accorder à une personne l'autorisation d'exécuter une instance Amazon MQ avec l'opération d'API `CreateBroker` Amazon MQ, vous incluez l'action `mq:CreateBroker` dans sa politique. Les déclarations de politique doivent inclure un élément `Action` ou `NotAction`. Amazon MQ définit son propre ensemble d'actions qui décrivent les tâches que vous pouvez effectuer avec ce service.

Pour spécifier plusieurs actions dans une seule déclaration, séparez-les par des virgules comme suit :

```
"Action": [
      "mq:action1",
      "mq:action2"
```

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `Describe`, incluez l'action suivante :

```
"Action": "mq:Describe*"
```

Pour afficher la liste des actions Amazon MQ, consultez Actions définies par Amazon MQ dans le Guide de l'utilisateur IAM.

Ressources

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Dans Amazon MQ, les AWS ressources principales sont un courtier de messages Amazon MQ et sa configuration. Les courtiers et configurations Amazon MQ sont chacun associés à des noms de ressources Amazon (ARNs) uniques, comme indiqué dans le tableau suivant.

Types de ressources	ARN	Clés de condition
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}

Pour plus d'informations sur le format de ARNs, consultez Amazon Resource Names (ARNs) et AWS Service Namespaces.

Par exemple, pour spécifier l'agent nommé `MyBroker` avec l'ID d'agent `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9` dans votre instruction, utilisez l'ARN suivant :

```
"Resource": "arn:aws:mq:us-
east-1:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9"
```

Pour spécifier tous les agents et configurations qui appartiennent à un compte spécifique, utilisez le caractère générique (*) :

```
"Resource": "arn:aws:mq:us-east-1:123456789012:*"
```

Certaines actions Amazon MQ, telles que celles destinées à la création de ressources, ne peuvent pas être exécutées sur une ressource spécifique. Dans ces cas-là, vous devez utiliser le caractère générique (*).

```
"Resource": "*"
```

L'action d'API `CreateTags` nécessite à la fois un agent et une configuration. Pour spécifier plusieurs ressources dans une seule instruction, séparez-les ARNs par des virgules.

```
"Resource": [
      "resource1",
      "resource2"
```

Pour consulter la liste des types de ressources Amazon MQ et leurs caractéristiques ARNs, consultez la section Ressources définies par Amazon MQ dans le guide de l'utilisateur IAM. Pour savoir les actions avec lesquelles vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par Amazon MQ.

Clés de condition

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` indique à quel moment les instructions s'exécutent en fonction de critères définis. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Amazon MQ ne fournit pas de clés de condition spécifiques au service, mais prend en charge l'utilisation de certaines clés de condition globales. Pour afficher une liste des clés de condition Amazon MQ, consultez le tableau ci-dessous ou Clés de condition pour Amazon MQ dans le Guide de l'utilisateur IAM. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez Actions définies par Amazon MQ.

Clés de condition	Description	Type
était : RequestTag /$ {} TagKey	Filtre les actions en fonction des balises qui sont transmises dans la demande.	String
était : ResourceTag /$ {} TagKey	Filtre les actions en fonction des balises associées à la ressource.	String
lois : TagKeys	Filtre les actions en fonction des clés de balise qui sont transmises dans la demande.	String

Exemples

Pour voir des exemples de politiques Amazon MQ basées sur l'identité, consultez Exemples de politique basée sur l'identité d'Amazon MQ.

Politiques basées sur des ressources Amazon MQ

Actuellement, Amazon MQ ne prend pas en charge l'authentification IAM à l'aide des autorisations basées sur les ressources ou des politiques basées sur les ressources.

Autorisation basée sur des balises Amazon MQ

Vous pouvez attacher des balises aux ressources Amazon MQ ou transmettre des balises dans une demande à Amazon MQ. Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `mq:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Amazon MQ prend en charge les politiques basées sur les balises. Par exemple, vous pouvez refuser l'accès aux ressources Amazon MQ qui incluent une balise avec la clé `environment` et la valeur`production` :

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "mq:DeleteBroker",
                "mq:RebootBroker",
                "mq:DeleteTags"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/environment": "production"
                }
            }
        }
    ]
}
```

Cette politique va `Deny` la possibilité de supprimer ou de redémarrer un agent Amazon MQ qui inclut la balise `environment/production`.

Pour plus d'informations sur le balisage, consultez :

- Ajouter des balises aux ressources Amazon MQ
- Contrôle de l'accès à l'aide de balises IAM

Rôles IAM Amazon MQ

Un rôle IAM est une entité de votre AWS compte qui possède des autorisations spécifiques.

Utilisation d'informations d'identification temporaires avec Amazon MQ

Vous pouvez utiliser des informations d’identification temporaires pour vous connecter à l’aide de la fédération, endosser un rôle IAM ou encore pour endosser un rôle intercompte. Vous obtenez des informations d'identification de sécurité temporaires en appelant des opérations d' AWS STS API telles que AssumeRoleou GetFederationToken.

Amazon MQ est compatible avec l'utilisation des informations d'identification temporaires.

Rôles du service

Cette fonction permet à un service d’endosser une fonction du service en votre nom. Ce rôle autorise le service à accéder à des ressources d’autres services pour effectuer une action en votre nom. Les rôles de service s’affichent dans votre compte IAM et sont la propriété du compte. Cela signifie qu’un administrateur IAM peut modifier les autorisations associées à ce rôle. Toutefois, une telle action peut perturber le bon fonctionnement du service.

Amazon MQ prend en charge les rôles de service.

Exemples de politique basée sur l'identité d'Amazon MQ

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier des ressources Amazon MQ. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l' AWS API AWS Management Console AWS CLI, ou. Un administrateur IAM doit créer des politiques IAM autorisant les utilisateurs et les rôles à exécuter des opérations d'API spécifiques sur les ressources spécifiées dont ils ont besoin. Il doit ensuite attacher ces politiques aux utilisateurs ou aux groupes IAM ayant besoin de ces autorisations.

Pour savoir comment créer une stratégie IAM basée sur l'identité à l'aide de ces exemples de documents de stratégie JSON, veuillez consulter Création de stratégies dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Rubriques

Bonnes pratiques en matière de politiques

Les stratégies basées sur l'identité déterminent si une personne peut créer, consulter ou supprimer des ressources Amazon MQ dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez politiques gérées par AWS ou politiques gérées par AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accordez les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation d'IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utilisez des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.
- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des

recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politiques avec IAM Access Analyzer dans le Guide de l'utilisateur IAM.

- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Sécurisation de l'accès aux API avec MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Utilisation de la console Amazon MQ

Pour accéder à la console Amazon MQ, vous devez disposer d'un ensemble minimum d'autorisations. Ces autorisations doivent vous permettre de répertorier et de consulter les informations relatives aux ressources Amazon MQ de votre AWS compte. Si vous créez une politique basée sur l'identité qui est plus restrictive que les autorisations minimales requises, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs et rôles IAM) tributaires de cette politique.

Pour vous assurer que ces entités peuvent toujours utiliser la console Amazon MQ, associez également la politique AWS gérée suivante aux entités. Pour en savoir plus, consultez Ajouter des autorisations à un utilisateur dans le guide de l'utilisateur IAM.

```
AmazonMQReadOnlyAccess
```

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API que vous tentez d'effectuer.

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or 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": "*"
        }
    ]
}
```

Authentification et autorisation d'API pour Amazon MQ

Amazon MQ utilise la signature de AWS demande standard pour l'authentification par API. Pour plus d'informations, consultez Signature des demandes d'API AWS dans le Références générales AWS.

> **Note**
>
> Actuellement, Amazon MQ ne prend pas en charge l'authentification IAM à l'aide des autorisations basées sur les ressources ou des politiques basées sur les ressources.

Pour autoriser AWS les utilisateurs à travailler avec des courtiers, des configurations et des utilisateurs, vous devez modifier les autorisations de votre politique IAM.

Rubriques

Autorisations IAM requises pour créer un agent Amazon MQ

Pour créer un agent, vous devez utiliser la politique IAM `AmazonMQFullAccess` ou inclure les autorisations EC2 suivantes dans votre politique IAM.

La politique personnalisée suivante est composée de deux déclarations (une conditionnelle) qui accordent des autorisations pour manipuler les ressources requises par Amazon MQ pour créer un agent ActiveMQ.

> **Important**
>
> - L'action `ec2:CreateNetworkInterface` est obligatoire pour permettre à Amazon MQ de créer une interface réseau Elastic (ENI) dans votre compte en votre nom.
> - L'action `ec2:CreateNetworkInterfacePermission` autorise Amazon MQ à attacher l'ENI à un agent ActiveMQ.
> - La clé de condition `ec2:AuthorizedService` s'assure que les autorisations d'ENI peuvent être accordées uniquement aux comptes de service 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"
            }
        }
    }]
}
```

Pour plus d'informations, consultez Étape 2 : créer un utilisateur et obtenir vos AWS informations d'identification et Ne jamais modifier ou supprimer l'interface réseau Elastic Amazon MQ.

Référence des autorisations d'API REST Amazon MQ

Le tableau suivant répertorie Amazon MQ REST APIs et les autorisations IAM correspondantes.

Amazon MQ REST APIs et autorisations requises

Amazon MQ REST APIs	Autorisations nécessaires
CreateBroker	mq:CreateBroker
CreateConfiguration	mq:CreateConfiguration
CreateTags	mq:CreateTags

Amazon MQ REST APIs	Autorisations nécessaires
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
ListTags	mq:ListTags
ListUsers	mq:ListUsers
RebootBroker	mq:RebootBroker
UpdateBroker	mq:UpdateBroker
UpdateConfiguration	mq:UpdateConfiguration
UpdateUser	mq:UpdateUser

Référence d'autorisations supplémentaires Amazon MQ

Le tableau suivant répertorie l'API Amazon MQ et les autorisations IAM supplémentaires requises pour des fonctionnalités spécifiques, telles que OAuth l'authentification 2.0.

API REST Amazon MQ	Autorisations	Description
UpdateBroker	`mq:UpdateBrokerAccessConfiguration`	Vous avez besoin de cette autorisation pour mettre à jour les options d'authentification et d'autorisation dans la configuration de broker associée. Pour de plus amples informations, veuillez consulter OAuth Authentification et autorisation 2.0 pour Amazon MQ pour RabbitMQ.

Autorisations au niveau des ressources pour les actions d'API Amazon MQ

Le terme autorisations au niveau des ressources font référence à la possibilité de spécifier les ressources sur lesquelles les utilisateurs sont autorisés à exécuter des actions. Amazon MQ prend partiellement en charge les autorisations au niveau des ressources. Pour certaines actions Amazon MQ, vous pouvez contrôler à quel moment les utilisateurs sont autorisés à utiliser ces actions en fonction des conditions qui doivent être satisfaites, ou les ressources spécifiques que les utilisateurs sont autorisés à utiliser.

Le tableau suivant décrit les actions d'API Amazon MQ qui prennent actuellement en charge les autorisations au niveau des ressources, ainsi que les ressources, les ressources et les clés de condition prises en charge pour chaque action. ARNs

> **Important**
>
> Si une action d'API Amazon MQ n'est pas répertoriée dans ce tableau, elle ne prend pas en charge les autorisations au niveau des ressources. Si une action d'API Amazon MQ ne prend pas en charge les autorisations au niveau des ressources, vous pouvez autoriser les utilisateurs à utiliser l'action, mais vous devez spécifier un caractère générique * pour l'élément ressource de votre déclaration de politique.

Action d'API	Types de ressource (*obligatoire)
CreateConfiguration	Configurations
CreateTags	agents, configurations
CreateUser	Agents
DeleteBroker	Agents
DeleteUser	Agents
DescribeBroker	Agents
DescribeConfiguration	Configurations
DescribeConfigurationRevision	Configurations
DescribeUser	Agents
ListConfigurationRevisions	Configurations
ListConfigurationRevisions	Configurations
ListTags	agents, configurations
ListUsers	Agents
RebootBroker	Agents
UpdateBroker	Agents
UpdateConfiguration	Configurations
UpdateUser	Agents

Authentification et autorisation du courtier

Amazon MQ propose différentes méthodes d'authentification et d'autorisation en fonction du type de moteur de votre broker.

Authentification et autorisation pour Amazon MQ pour ActiveMQ

Amazon MQ pour ActiveMQ prend en charge les méthodes d'authentification et d'autorisation suivantes :

Authentification et autorisation simples

Dans cette méthode, les utilisateurs du broker sont créés et gérés via la console ou l'API Amazon MQ. Les utilisateurs peuvent être configurés avec des autorisations spécifiques pour accéder aux files d'attente, aux rubriques et à la console Web ActiveMQ. Pour plus d'informations sur cette méthode, consultez la section Création d'un utilisateur de courtier ActiveMQ.

Authentification et autorisation LDAP

Dans cette méthode, les utilisateurs du broker s'authentifient à l'aide des informations d'identification stockées sur votre serveur LDAP. Vous pouvez ajouter, supprimer et modifier des utilisateurs et attribuer des autorisations aux sujets et aux files d'attente via le serveur LDAP, en fournissant une authentification et une autorisation centralisées. Pour plus d'informations sur cette méthode, consultez la section Intégration des courtiers ActiveMQ à LDAP.

Authentification et autorisation pour Amazon MQ pour RabbitMQ

Amazon MQ pour RabbitMQ prend en charge les méthodes d'authentification et d'autorisation suivantes :

Authentification et autorisation simples

Dans cette méthode, les utilisateurs du courtier sont stockés en interne dans le courtier RabbitMQ et gérés via la console Web ou l'API de gestion. Les autorisations pour les hôtes virtuels, les échanges, les files d'attente et les sujets sont configurées directement dans RabbitMQ. Il s'agit de la méthode par défaut. Pour plus d'informations, consultez Authentification et autorisation simples.

OAuth Authentification et autorisation 2.0

Dans cette méthode, les utilisateurs du broker et leurs autorisations sont gérés par un fournisseur d'identité OAuth 2.0 externe (IdP). L'authentification des utilisateurs et les autorisations de ressources

pour les hôtes virtuels, les échanges, les files d'attente et les sujets sont centralisées via le système de périmètre du fournisseur OAuth 2.0. Cela simplifie la gestion des utilisateurs et permet l'intégration aux systèmes d'identité existants. Pour plus d'informations, voir Authentification et autorisation OAuth 2.0.

Authentification et autorisation IAM

Dans cette méthode, les utilisateurs du broker s'authentifient à l'aide des informations d'identification AWS IAM via la fédération sortante IAM. Les informations d'identification IAM sont utilisées pour obtenir des jetons JWT auprès du AWS Security Token Service (STS), et ces jetons JWT servent de jetons OAuth 2.0 pour l'authentification. Cette méthode s'appuie sur le support OAuth 2.0 existant dans Amazon MQ pour RabbitMQ, qui agit en tant que fournisseur d' AWS identité 2.0. OAuth L'authentification des utilisateurs est gérée par AWS IAM, tandis que les autorisations de ressources pour les hôtes virtuels, les échanges, les files d'attente et les sujets sont gérées via des politiques IAM et des alias de portée configurés dans RabbitMQ. Pour plus d'informations, consultez Authentification et autorisation IAM.

Authentification et autorisation LDAP

Dans cette méthode, les utilisateurs du broker et leurs autorisations sont gérés par un service d'annuaire LDAP externe. L'authentification des utilisateurs et les autorisations de ressources sont centralisées via le serveur LDAP, ce qui permet aux utilisateurs d'accéder à RabbitMQ en utilisant leurs informations d'identification de service d'annuaire existantes. Pour plus d'informations, consultez Authentification et autorisation LDAP.

Authentification et autorisation HTTP

Dans cette méthode, les utilisateurs du broker et leurs autorisations sont gérés par un serveur HTTP externe. L'authentification des utilisateurs et les autorisations de ressources sont centralisées via le serveur HTTP, ce qui permet aux utilisateurs d'accéder à RabbitMQ en utilisant leur propre fournisseur d'authentification et d'autorisation. Pour plus d'informations sur cette méthode, consultez Authentification et autorisation HTTP.

Authentification par certificat SSL

Amazon MQ prend en charge le protocole TLS mutuel (MTL) pour les courtiers RabbitMQ. Le plugin d'authentification SSL utilise des certificats clients issus de connexions mTLS pour authentifier les utilisateurs. Dans cette méthode, les utilisateurs du broker sont authentifiés à l'aide de certificats clients X.509 au lieu de leur nom d'utilisateur et de leur mot de passe. Le certificat du client est validé

auprès d'une autorité de certification (CA) fiable, et le nom d'utilisateur est extrait d'un champ du certificat, tel que le nom commun (CN) ou le nom alternatif du sujet (SAN). Cette méthode fournit une authentification forte sans transmettre d'informations d'identification sur le réseau. Pour plus d'informations, consultez la section Authentification par certificat SSL.

> **Note**
>
> RabbitMQ prend en charge plusieurs méthodes d'authentification et d'autorisation à utiliser simultanément. Par exemple, vous pouvez activer à la fois l'authentification OAuth 2.0 et l'authentification simple (interne). Pour plus d'informations, consultez la section du didacticiel OAuth 2.0 sur l'activation à la fois de l'authentification OAuth 2.0 et de l'authentification simple (interne) et la documentation sur le contrôle d'accès RabbitMQ.
> Amazon MQ recommande de créer un utilisateur interne lors du test des configurations d'authentification. Cela permet de valider la configuration des accès à l'aide de l'API de gestion RabbitMQ. Pour plus d'informations, consultez la section Validation des accès.

AWS politiques gérées pour Amazon MQ

Une politique AWS gérée est une politique autonome créée et administrée par AWS. AWS les politiques gérées sont conçues pour fournir des autorisations pour de nombreux cas d'utilisation courants afin que vous puissiez commencer à attribuer des autorisations aux utilisateurs, aux groupes et aux rôles.

N'oubliez pas que les politiques AWS gérées peuvent ne pas accorder d'autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles sont accessibles à tous les AWS clients. Nous vous recommandons de réduire encore les autorisations en définissant des politiques gérées par le client qui sont propres à vos cas d'utilisation.

Vous ne pouvez pas modifier les autorisations définies dans les politiques AWS gérées. Si les autorisations définies dans une politique AWS gérée sont AWS mises à jour, la mise à jour affecte toutes les identités principales (utilisateurs, groupes et rôles) auxquelles la politique est attachée. AWS est le plus susceptible de mettre à jour une politique AWS gérée lorsqu'une nouvelle Service AWS est lancée ou lorsque de nouvelles opérations d'API sont disponibles pour les services existants.

Pour plus d'informations, consultez Politiques gérées par AWS dans le Guide de l'utilisateur IAM.

Amazon MQ prend en charge les politiques AWS gérées suivantes :

- AmazonMQApiFullAccess
- AmazonMQApiReadOnlyAccess
- Amazon MQFull Access
- AmazonMQReadOnlyAccess
- AmazonMQServiceRolePolicy

AWS politique gérée : Amazon MQService RolePolicy

Vous ne pouvez pas attacher `AmazonMQServiceRolePolicy` à vos entités IAM. Cette politique est attachée à un rôle lié à un service qui permet à Amazon MQ de réaliser des actions en votre nom. Pour plus d'informations sur cette politique d'autorisation et sur les actions qu'elle permet à Amazon MQ d'effectuer, consultez the section called “Autorisations du rôle lié à un service pour Amazon MQ”.

Amazon MQ met à jour les politiques gérées AWS

Consultez les informations relatives aux mises à jour des politiques AWS gérées pour Amazon MQ depuis que ce service a commencé à suivre ces modifications. Pour recevoir des alertes automatiques sur les modifications apportées à cette page, abonnez-vous au flux RSS sur la page de l'historique des documents Amazon MQ.

Modifier	Description	Date
Amazon MQ a commencé à assurer le suivi des modifications	Amazon MQ a commencé à suivre les modifications apportées à ses politiques AWS gérées.	5 mai 2021

Utilisation des rôles liés à un service pour Amazon MQ

Amazon MQ utilise des rôles liés à un Gestion des identités et des accès AWS service (IAM). Un rôle lié à un service est un type unique de rôle IAM lié directement à Amazon MQ. Les rôles liés à un

service sont prédéfinis par Amazon MQ et incluent toutes les autorisations requises par le service pour appeler AWS d'autres services en votre nom.

Un rôle lié à un service simplifie la configuration d'Amazon MQ, car vous n'avez pas besoin d'ajouter manuellement les autorisations requises. Amazon MQ définit les autorisations de ses rôles liés à un service ; sauf définition contraire, seul Amazon MQ peut endosser ses rôles. Les autorisations définies comprennent la politique de confiance et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous pouvez supprimer un rôle lié à un service uniquement après la suppression préalable de ses ressources connexes. Vos ressources Amazon MQ sont ainsi protégées, car vous ne pouvez pas involontairement supprimer l'autorisation d'accéder aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM et recherchez les services avec un Oui dans la colonne Rôle lié à un service. Choisissez un Oui ayant un lien permettant de consulter les détails du rôle pour ce service.

Autorisations du rôle lié à un service pour Amazon MQ

Amazon MQ utilise le rôle lié au service nommé MQ AWSServiceRoleForAmazon— Amazon MQ utilise ce rôle lié au service pour appeler les services en votre nom. AWS

Le rôle lié au service AWSService RoleForAmazon MQ fait confiance aux services suivants pour assumer le rôle :

- `mq.amazonaws.com`

Amazon MQ utilise la politique d'autorisation `AmazonMQServiceRolePolicy`, qui est attachée au rôle lié au service AWSService RoleForAmazon MQ, pour effectuer les actions suivantes sur les ressources spécifiées :

- Action : `ec2:CreateVpcEndpoint` sur la ressource `vpc`.
- Action : `ec2:CreateVpcEndpoint` sur la ressource `subnet`.
- Action : `ec2:CreateVpcEndpoint` sur la ressource `security-group`.
- Action : `ec2:CreateVpcEndpoint` sur la ressource `vpc-endpoint`.

- Action : `ec2:DescribeVpcEndpoints` sur la ressource `vpc`.
- Action : `ec2:DescribeVpcEndpoints` sur la ressource `subnet`.
- Action : `ec2:CreateTags` sur la ressource `vpc-endpoint`.
- Action : `logs:PutLogEvents` sur la ressource `log-group`.
- Action : `logs:DescribeLogStreams` sur la ressource `log-group`.
- Action : `logs:DescribeLogGroups` sur la ressource `log-group`.
- Action : `CreateLogStream` sur la ressource `log-group`.
- Action : `CreateLogGroup` sur la ressource `log-group`.

Lorsque vous créez un agent Amazon MQ for RabbitMQ, la politique d'autorisation `AmazonMQServiceRolePolicy` permet à Amazon MQ d'effectuer les tâches suivantes en votre nom.

- Créez un point de terminaison Amazon VPC pour l'agent à l'aide du VPC Amazon, du sous-réseau et du groupe de sécurité que vous fournissez. Vous pouvez utiliser le point de terminaison créé pour votre agent pour vous connecter à l'agent via la console de gestion RabbitMQ, l'API de gestion ou par programmation.
- Créez des groupes de journaux et publiez les journaux des courtiers sur 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/*"
            ]
        }
    ]
}
```

Vous devez configurer les autorisations de manière à permettre à une entité IAM (comme un utilisateur, un groupe ou un rôle) de créer, modifier ou supprimer un rôle lié à un service. Pour en savoir plus, consultez Service-Linked Role Permissions (autorisations du rôle lié à un service) dans le Guide de l'utilisateur IAM.

Création d'un rôle lié à un service pour Amazon MQ

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous créez un courtier pour la première fois, Amazon MQ crée un rôle lié à un service pour appeler les AWS services en votre nom. Tous les agents suivants que vous créez utiliseront le même rôle et aucun nouveau rôle n'est créé.

> **Important**
>
> Ce rôle lié à un service peut apparaître dans votre compte si vous avez effectué une action dans un autre service qui utilise les fonctions prises en charge par ce rôle. Pour de plus amples informations, veuillez consulter Un nouveau rôle est apparu dans mon compte IAM.

Si vous supprimez ce rôle lié à un service et que vous avez ensuite besoin de le recréer, vous pouvez utiliser la même procédure pour recréer le rôle dans votre compte.

Vous pouvez également utiliser la console IAM pour créer un rôle lié au service avec le cas d'utilisation Amazon MQ. Dans l'API AWS CLI ou dans l' AWS API, créez un rôle lié à un service avec le nom du `mq.amazonaws.com` service. Pour plus d'informations, consultez Création d'un rôle lié à un service dans le Guide de l'utilisateur IAM. Si vous supprimez ce rôle lié à un service, vous pouvez utiliser ce même processus pour créer le rôle à nouveau.

> **Important**
>
> Les rôles liés à un service sont créés uniquement pour Amazon MQ pour RabbitMQ.

Modification d'un rôle lié à un service pour Amazon MQ

Amazon MQ ne vous permet pas de modifier le rôle lié au service AWSService RoleForAmazon MQ. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour plus d'informations, consultez Modification d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Suppression d'un rôle lié à un service pour Amazon MQ

Si vous n'avez plus besoin d'utiliser une fonctionnalité ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. De cette façon, vous n'avez aucune entité inutilisée qui n'est pas surveillée ou gérée activement. Cependant, vous devez nettoyer les ressources de votre rôle lié à un service avant de pouvoir les supprimer manuellement.

> **Note**
>
> Si le service Amazon MQ utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression peut échouer. Si cela se produit, patientez quelques minutes et réessayez.

Pour supprimer les ressources Amazon MQ utilisées par le MQ AWSService RoleForAmazon

- Supprimez vos courtiers Amazon MQ à l'aide de la AWS Management Console CLI Amazon MQ ou de l'API Amazon MQ. Pour de plus amples informations sur la suppression d'un agent, veuillez consulter ???.

Pour supprimer manuellement le rôle lié au service à l'aide d'IAM

Utilisez la console IAM AWS CLI, le ou l' AWS API pour supprimer le rôle lié au service AWSService RoleForAmazon MQ. Pour en savoir plus, consultez Suppression d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Régions prises en charge pour les rôles liés à un service Amazon MQ

Amazon MQ prend en charge l'utilisation des rôles liés à un service dans toutes les régions où le service est disponible. Pour de plus amples informations, consultez Regions and Endpoints AWS (Régions et points de terminaison) .

Nom de la région	Identité de la région	Prise en charge dans Amazon MQ
USA Est (Virginie du Nord)	us-east-1	Oui
USA Est (Ohio)	us-east-2	Oui
USA Ouest (Californie du Nord)	us-west-1	Oui
USA Ouest (Oregon)	us-west-2	Oui
Asie-Pacifique (Mumbai)	ap-south-1	Oui
Asie-Pacifique (Osaka)	ap-northeast-3	Oui
Asie-Pacifique (Séoul)	ap-northeast-2	Oui
Asie-Pacifique (Singapour)	ap-southeast-1	Oui
Asie-Pacifique (Sydney)	ap-southeast-2	Oui
Asie-Pacifique (Tokyo)	ap-northeast-1	Oui
Canada (Centre)	ca-central-1	Oui
Europe (Francfort)	eu-central-1	Oui
Europe (Irlande)	eu-west-1	Oui

Nom de la région	Identité de la région	Prise en charge dans Amazon MQ
Europe (Londres)	eu-west-2	Oui
Europe (Paris)	eu-west-3	Oui
Amérique du Sud (São Paulo)	sa-east-1	Oui
AWS GovCloud (US)	us-gov-west-1	Non

Résolution de problèmes pour identité et accès Amazon MQ

Utilisez les informations suivantes pour identifier et résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec Amazon MQ et IAM.

Rubriques

- Action à effectuer dans Amazon MQ refusée
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes ressources Amazon MQ

Action à effectuer dans Amazon MQ refusée

S'il vous AWS Management Console indique que vous n'êtes pas autorisé à effectuer une action, vous devez contacter votre administrateur pour obtenir de l'aide. Votre administrateur est la personne qui vous a fourni vos informations de connexion.

L'exemple d'erreur suivant se produit lorsque l'`mateojackson`utilisateur essaie d'utiliser la console pour afficher les détails d'un *`widget`* mais ne dispose pas des `mq:`*`GetWidget`* autorisations nécessaires.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 mq:GetWidget on resource: my-example-widget
```

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses politiques pour lui permettre d'accéder à la ressource *`my-example-widget`* à l'aide de l'action `mq:`*`GetWidget`*.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter l'action `iam:PassRole`, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à Amazon MQ.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, vous devez disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans Amazon MQ. Toutefois, l'action nécessite que le service ait des autorisations accordées par une fonction du service. Mary n'est pas autorisée à transmettre le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d’exécuter l’action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d’identification de connexion.

Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes ressources Amazon MQ

Vous pouvez créer un rôle que les utilisateurs provenant d’autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour plus d’informations, consultez les éléments suivants :

- Pour savoir si Amazon MQ est compatible avec ces fonctionnalités, veuillez consulter Fonctionnement d'Amazon MQ avec IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.

- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour en savoir plus sur la différence entre l'utilisation des rôles et des politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Validation de conformité pour Amazon MQ

Des auditeurs tiers évaluent la sécurité et la conformité d'Amazon MQ dans le cadre de plusieurs programmes de AWS conformité. Il s'agit notamment des certifications SOC, PCI, HIPAA.

Pour savoir si un programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de AWS conformité Programmes AWS de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. Pour plus d'informations sur votre responsabilité en matière de conformité lors de l'utilisation Services AWS, consultez AWS la documentation de sécurité.

Résilience dans Amazon MQ

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section Infrastructure AWS mondiale.

Sécurité de l'infrastructure dans Amazon MQ

En tant que service géré, Amazon MQ est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section Sécurité du AWS cloud. Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section Protection de l'infrastructure dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à Amazon MQ via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

Bonnes pratiques de sécurité pour Amazon MQ

Les modèles de conception suivants peuvent améliorer la sécurité de votre agent Amazon MQ.

Rubriques

- Préférer les agents sans accessibilité publique
- Toujours configurer un plan d'autorisation
- Bloquer les protocoles inutiles avec des groupes de sécurité VPC

Pour plus d'informations sur la façon dont Amazon MQ chiffre vos données, ainsi que sur la liste des protocoles pris en charge, consultez Protection des données.

Préférer les agents sans accessibilité publique

Les agents créés sans accessibilité publique ne sont pas accessibles depuis l'extérieur de votre VPC. Cela réduit considérablement la vulnérabilité de votre courtier aux attaques par déni de service (DDoS) distribué provenant de l'Internet public. Pour plus d'informations, consultez Comment vous

préparer aux attaques DDo S en réduisant votre surface d'attaque sur le blog consacré à la AWS sécurité.

Toujours configurer un plan d'autorisation

Étant donné qu'aucun plan d'autorisation n'est configuré pour ActiveMQ par défaut, tout utilisateur authentifié peut effectuer n'importe quelle action sur l'agent. Ainsi, une bonne pratique consiste à limiter les autorisations par groupe. Pour de plus amples informations, veuillez consulter `authorizationEntry`.

> **Important**
>
> Si vous spécifiez un plan d'autorisation qui n'inclut pas le groupe `activemq-webconsole`, vous ne pouvez pas utiliser la console web ActiveMQ car le groupe n'est pas autorisé à envoyer des messages à l'agent Amazon MQ ou à recevoir des messages de ce dernier.

Bloquer les protocoles inutiles avec des groupes de sécurité VPC

Pour améliorer la sécurité des courtiers privés, vous devez limiter les connexions de protocoles et de ports inutiles en configurant correctement votre groupe de sécurité Amazon VPC. Par exemple, pour restreindre l'accès à la plupart des protocoles tout en autorisant l'accès à la console Web OpenWire et à celle-ci, vous pouvez autoriser l'accès uniquement aux protocoles 61617 et 8162. Cela limite votre exposition en bloquant les protocoles que vous n'utilisez pas, tout en permettant OpenWire à la console Web de fonctionner normalement.

Autorisez uniquement les ports de protocole que vous utilisez.

- AMQP : 5671
- MQTT : 8883
- OpenWire: 61617
- STOMP : 61614
- WebSocket: 61619

Pour plus d'informations, consultez :

- Groupes de sécurité pour votre VPC

- Groupe de sécurité par défaut pour votre VPC
- Utilisation des groupes de sécurité

Journalisation et surveillance d’agents Amazon MQ

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances de vos AWS solutions. Vous devez collecter des données de surveillance provenant de toutes les parties de votre AWS solution afin de pouvoir corriger plus facilement une défaillance multipoint, le cas échéant. AWS fournit plusieurs outils pour surveiller les ressources de votre Amazon MQ et répondre aux incidents potentiels :

Vous pouvez l'utiliser CloudWatch pour consulter et analyser les statistiques de votre courtier Amazon MQ. Vous pouvez consulter et analyser les statistiques de votre courtier depuis la CloudWatch console, le AWS CLI, ou le CloudWatch AWS CLI. CloudWatch les métriques d'Amazon MQ sont automatiquement interrogées par le courtier, puis reportées à CloudWatch chaque minute. Pour les courtiers ActiveMQ CloudWatch , surveille uniquement les 1 000 premières destinations. Pour les courtiers RabbitMQ, CloudWatch surveille uniquement les 500 premières destinations, classées par nombre de consommateurs.

Pour obtenir la liste complète de toutes les métriques Amazon MQ, consultez CloudWatch Mesures disponibles : Amazon MQ pour les courtiers ActiveMQ.

Pour plus d'informations sur la création CloudWatch d'une alarme pour une métrique, consultez la section Créer ou modifier une CloudWatch alarme dans le guide de CloudWatch l'utilisateur Amazon.

Accès aux CloudWatch métriques pour Amazon MQ

Vous pouvez accéder aux CloudWatch métriques à l'aide de l'API AWS Management Console AWS CLI,, et.

Vous souhaiterez peut-être accéder aux CloudWatch métriques sans utiliser le AWS Management Console.

Pour accéder aux métriques Amazon MQ à l'aide de AWS CLI, utilisez la `get-metric-statistics` commande. Pour plus d'informations, consultez Obtenir des statistiques pour une métrique dans le guide de CloudWatch l'utilisateur Amazon.

Pour accéder aux métriques Amazon MQ à l'aide de l' CloudWatch API, utilisez l'`GetMetricStatistics`action. Pour plus d'informations, consultez Obtenir des statistiques pour une métrique dans le guide de CloudWatch l'utilisateur Amazon.

Accès aux CloudWatch métriques à l'aide du AWS Management Console

L'exemple suivant vous montre comment accéder aux CloudWatch métriques d'Amazon MQ à l'aide du AWS Management Console.Si vous êtes déjà connecté à la console Amazon MQ, sur la page des informations du courtier, choisissez Actions, Afficher les métriques. CloudWatch

1. Connectez-vous à la console CloudWatch.
2. Dans le volet de navigation, choisissez Métriques.
3. Sélectionnez l'espace de nom de métrique AmazonMQ.
4. Sélectionnez l'une des dimensions de métrique suivantes :

   - Métriques d'agent
   - Métriques de file d'attente par agent
   - Métriques de rubrique par agent

   Dans cet exemple, Broker Metrics (Métriques d'agent) est sélectionné.
5. Vous pouvez désormais examiner vos métriques Amazon MQ :

   - Pour trier les métriques, utilisez l’en-tête de colonne.
   - Cochez la case en regard d'une métrique pour la représenter graphiquement.
   - Pour filtrer par métrique, choisissez le nom de la métrique, puis Add to search (Ajouter à la recherche).

CloudWatch Mesures disponibles : Amazon MQ pour les courtiers ActiveMQ

Mesures Amazon MQ pour ActiveMQ

Mesure	Unité	Description
`AmqpMaximumConnections`	Nombre	Le nombre maximal de clients que vous pouvez connecter à votre agent à

Mesure	Unité	Description
		l'aide d'AMQP. Pour de plus amples informations sur les quotas de connexion, veuillez consulter Quotas in Amazon MQ.
`BurstBalance`	Pourcentage	Pourcentage de crédits en rafale restant sur le volume Amazon EBS permettant de conserver les données de message pour les agents optimisés pour le débit. Si ce solde atteint zéro, les IOPS fournies par le volume Amazon EBS diminueront jusqu'à ce que le solde en rafale soit rechargé. Pour de plus amples informations sur le fonctionnement des soldes en rafale dans Amazon EBS, consultez Crédits I/O et performances en rafale.

Mesure	Unité	Description
`CpuCreditBalance`	Crédits (minutes vCPU)	**Important** Cette mesure est disponible uniquement pour le type d'instance d'agent `mq.t2.micro` . Les mesures de crédits UC sont disponibles à des intervalles de 5 minutes. Nombre de crédits UC gagnés qu'une instance a accumulés depuis son lancement ou son démarrage (y compris le nombre de crédits de lancement). L'instance d'agent peut dépenser les crédits figurant dans le solde de crédits pour dépasser le niveau de base de l'utilisation de l'UC. Les crédits sont accumulés dans le solde de crédits quand ils sont gagnés et supprimés du solde de crédits lorsqu'ils sont dépensés. Le solde de crédits présente une limite maximale. Une fois que la limite est atteinte, les

Mesure	Unité	Description
		nouveaux crédits gagnés sont rejetés.
CpuUtilization	Pourcentage	Pourcentage d'unités de calcul Amazon EC2 allouées actuellement utilisées par l'agent.
CurrentConnectionsCount	Nombre	Nombre actuel de connexions actives sur l'agent actuel.
EstablishedConnectionsCount	Nombre	Nombre total de connexions, actives et inactives, qui ont été établies sur l'agent.
HeapUsage	Pourcentage	Pourcentage de la limite de mémoire JVM ActiveMQ actuellement utilisé par l'agent.
InactiveDurableTopicSubscribersCount	Nombre	Nombre d'abonnés à une rubrique durable inactifs, jusqu'à 2 000 maximum.
JobSchedulerStorePercentUsage	Pourcentage	Pourcentage d'espace disque utilisé par le magasin du planificateur de tâches.
JournalFilesForFastRecovery	Nombre	Nombre de fichiers journaux qui seront réutilisés après un arrêt normal.
JournalFilesForFullRecovery	Nombre	Nombre de fichiers journaux qui seront réutilisés après un arrêt incorrect.

Mesure	Unité	Description
`MqttMaximumConnections`	Nombre	Le nombre maximal de clients que vous pouvez connecter à votre agent à l'aide de MQTT. Pour de plus amples informations sur les quotas de connexion, veuillez consulter Quotas in Amazon MQ.
`NetworkConnectorConnectionCount`	Nombre	Nombre de nœuds connectés au courtier dans un réseau de courtiers utilisant NetworkConnector.
`NetworkIn`	Octets	Volume de trafic entrant pour l'agent.
`NetworkOut`	Octets	Volume de trafic sortant pour l'agent.
`OpenTransactionCount`	Nombre	Nombre total de transactions en cours.
`OpenwireMaximumConnections`	Nombre	Le nombre maximum de clients que vous pouvez utiliser pour vous connecter à votre courtier OpenWire. Pour de plus amples informations sur les quotas de connexion, veuillez consulter Quotas in Amazon MQ.

Mesure	Unité	Description
StompMaximumConnections	Nombre	Le nombre maximal de clients que vous pouvez connecter à votre agent à l'aide de STOMP. Pour de plus amples informations sur les quotas de connexion, veuillez consulter Quotas in Amazon MQ.
StorePercentUsage	Pourcentage	Pourcentage utilisé par la limite de stockage. Si ce pourcentage atteint 100 %, l'agent refusera les messages.
TempPercentUsage	Pourcentage	Pourcentage de stockage temporaire disponible utilisé par les messages non persistants.
TotalConsumerCount	Nombre	Nombre de consommateurs de message abonnés à des destinations sur l'agent actuel.
TotalMessageCount	Nombre	Nombre de messages stockés sur l'agent.
TotalProducerCount	Nombre	Nombre de producteurs de message actifs dans des destinations sur l'agent actuel.
VolumeReadOps	Nombre	Nombre d'opérations de lecture effectuées sur le volume Amazon EBS.
VolumeWriteOps	Nombre	Nombre d'opérations d'écriture effectuées sur le volume Amazon EBS.

Mesure	Unité	Description
`WsMaximumConnections`	Nombre	Le nombre maximum de clients que vous pouvez utiliser pour vous connecter à votre courtier WebSocket . Pour de plus amples informations sur les quotas de connexion, veuillez consulter Quotas in Amazon MQ.

Dimensions pour les mesures de l'agent ActiveMQ

Dimension	Description
`Broker`	Nom de l'agent **Note** Un agent à instance unique comporte le suffixe -1. Un active/standby broker de haute disponibilité possède les suffixes -1 et -2 pour sa paire redondante.

Mesures de destination ActiveMQ (file d'attente et rubrique)

Important

Les statistiques suivantes incluent le nombre de minutes pour la période de CloudWatch sondage.

- `EnqueueCount`
- `ExpiredCount`
- `DequeueCount`
- `DispatchCount`

- `InFlightCount`

Par exemple, dans une période CloudWatch de cinq minutes,`EnqueueCount` a cinq valeurs de comptage, soit une par tranche d'une minute de la période. Les statistiques `Minimum` et `Maximum` fournissent les valeurs la plus basse et la plus élevée par minute au cours de la période spécifiée.

Mesure	Unité	Description
`ConsumerCount`	Nombre	Nombre de consommateurs abonnés à la destination.
`EnqueueCount`	Nombre	Nombre de messages envoyés à la destination par minute.
`EnqueueTime`	Durée (millisecondes)	Le end-to-end temps de latence entre le moment où un message arrive chez un courtier et celui où il est livré au consommateur. **Note** `EnqueueTime` ne mesure pas la end-to-end latence entre le moment où un message est envoyé par un producteur et celui où il parvient au courtier, ni la latence entre le moment où un message est reçu par un courtier et celui où il en est

Mesure	Unité	Description
		accusé réception par le courtier. Au contraire, `EnqueueTime` est le nombre de millisecondes à partir du moment où un message est reçu par l'agent jusqu'à ce qu'il soit livré avec succès à un consommateur.
`ExpiredCount`	Nombre	Nombre de messages qui n'ont pas pu être transmis car ils ont expiré, par minute.
`DispatchCount`	Nombre	Nombre de messages envoyés à des consommateurs par minute.
`DequeueCount`	Nombre	Nombre de messages reconnus par des consommateurs par minute.
`InFlightCount`	Nombre	Nombre de messages envoyés aux consommateurs qui n'ont pas été reconnus.
`ReceiveCount`	Nombre	Nombre de messages qui ont été reçus de l'agent à distance pour un connecteur de réseau duplex.
`MemoryUsage`	Pourcentage	Pourcentage de la limite de mémoire actuellement utilisée par la destination.

Mesure	Unité	Description
`ProducerCount`	Nombre	Nombre de producteurs pour la destination.
`QueueSize`	Nombre	Nombre de messages dans le file d'attente. **Important** Cette mesure s'appliqu e uniquement aux files d'attente.
`TotalEnqueueCount`	Nombre	Nombre total de messages envoyés à l'agent.
`TotalDequeueCount`	Nombre	Nombre total de messages consommés par les clients.

Note

Les mesures `TotalEnqueueCount` et `TotalDequeueCount` incluent des messages pour les rubriques consultatives. Pour plus d'informations sur les messages de rubrique consultative, consultez la documentation ActiveMQ.

Dimensions pour les mesures de destination ActiveMQ (file d'attente et rubrique)

Dimension	Description
`Broker`	Nom de l'agent. **Note** Un agent à instance unique comporte le suffixe `-1`. Un active/standby courtier

Dimension	Description
	pour la haute disponibilité possède les suffixes `-1` et sa `-2` paire redondante.
`Topic` ou `Queue`	Nom de la rubrique ou de la file d'attente.
`NetworkConnector`	Nom du connecteur de réseau.

CloudWatch Mesures disponibles pour Amazon MQ pour les courtiers RabbitMQ

Mesures d'agent RabbitMQ

Mesure	Unité	Description
`ExchangeCount`	Nombre	Nombre total d'échanges configurés sur l'agent.
`QueueCount`	Nombre	Nombre total de files d'attente configurées sur l'agent.
`ConnectionCount`	Nombre	Nombre total de connexions établies sur l'agent.
`ChannelCount`	Nombre	Nombre total de canaux établis sur l'agent.
`ConsumerCount`	Nombre	Nombre total de consommateurs connectés à l'agent.
`MessageCount`	Nombre	Nombre de messages dans les files d'attente.

Mesure	Unité	Description
		Note Le nombre produit est la somme totale des messages prêts et non reconnus sur l'agent.
`MessageReadyCount`	Nombre	Nombre de messages prêts dans les files d'attente.
`MessageUnacknowledgedCount`	Nombre	Nombre de messages non reconnus dans les files d'attente.
`PublishRate`	Nombre	Fréquence de publication des messages sur l'agent. Le nombre produit représente le nombre de messages par seconde au moment de l'échantillonnage.
`ConfirmRate`	Nombre	Fréquence à laquelle le serveur RabbitMQ confirme les messages publiés. Vous pouvez comparer cette mesure avec `PublishRate` pour mieux comprendre les performances de votre agent. Le nombre produit représente le nombre de messages par seconde au moment de l'échantillonnage.

Mesure	Unité	Description
`AckRate`	Nombre	Fréquence à laquelle les messages sont confirmés par les consommateurs. Le nombre produit représente le nombre de messages par seconde au moment de l'échantillonnage.
`SystemCpuUtilization`	Pourcentage	Pourcentage d’unités de calcul Amazon EC2 allouées actuellement utilisées par l’agent. Pour les déploiements en cluster, cette valeur représente l'agrégat des valeurs de métrique correspondantes des trois nœuds RabbitMQ.
`RabbitMQMemLimit`	Octets	La limite de RAM pour un agent RabbitMQ. Pour les déploiements en cluster, cette valeur représente l'agrégat des valeurs de métrique correspondantes des trois nœuds RabbitMQ.
`RabbitMQMemUsed`	Octets	Le volume de RAM utilisé par un agent RabbitMQ. Pour les déploiements en cluster, cette valeur représente l'agrégat des valeurs de métrique correspondantes des trois nœuds RabbitMQ.

Mesure	Unité	Description
`RabbitMQDiskFreeLimit`	Octets	La limite de disque pour un agent RabbitMQ. Pour les déploiements en cluster, cette valeur représente l'agrégat des valeurs de métrique correspondantes des trois nœuds RabbitMQ. Cette mesure est différente selon la taille d'instance.
`RabbitMQDiskFree`	Octets	Le volume total d'espace disque disponible dans un agent RabbitMQ. Lorsque l'utilisation du disque dépasse sa limite, le cluster bloque toutes les connexions du producteur. Pour les déploiements en cluster, cette valeur représente l'agrégat des valeurs de métrique correspondantes des trois nœuds RabbitMQ.
`RabbitMQFdUsed`	Nombre	Nombre de descripteurs de fichiers utilisés. Pour les déploiements en cluster, cette valeur représente l'agrégat des valeurs de métrique correspondantes des trois nœuds RabbitMQ.

Mesure	Unité	Description
`RabbitMQIOReadAverageTime`	Nombre	Temps moyen (en millisecondes) nécessaire à RabbitMQ pour effectuer une opération de lecture. La valeur est proportionnelle à la taille du message.
`RabbitMQIOWriteAverageTime`	Nombre	Temps moyen (en millisecondes) nécessaire à RabbitMQ pour effectuer une opération d'écriture. La valeur est proportionnelle à la taille du message.

Dimensions pour les mesures de l'agent RabbitMQ

Dimension	Description
`Broker`	Nom de l'agent.

Mesures du nœud RabbitMQ

Mesure	Unité	Description
`SystemCpuUtilization`	Pourcentage	Pourcentage d'unités de calcul Amazon EC2 allouées actuellement utilisées par l'agent.
`RabbitMQMemLimit`	Octets	Limite de RAM pour un nœud RabbitMQ.
`RabbitMQMemUsed`	Octets	Volume de RAM utilisé par un nœud RabbitMQ. Lorsque

Mesure	Unité	Description
		l'utilisation de la mémoire dépasse la limite, le cluster bloque toutes les connexions du producteur.
`RabbitMQDiskFreeLimit`	Octets	Limite de disque pour un nœud RabbitMQ. Cette mesure est différente selon la taille d'instance.
`RabbitMQDiskFree`	Octets	Volume total d'espace disque disponible dans un nœud RabbitMQ. Lorsque l'utilisation du disque dépasse sa limite, le cluster bloque toutes les connexions du producteur.
`RabbitMQFdUsed`	Nombre	Nombre de descripteurs de fichiers utilisés.

Dimensions pour les mesures du nœud RabbitMQ

Dimension	Description
`Node`	Le nom du nœud. **Note** Un nom de nœud se compose de deux parties : un préfixe (habituellement `rabbit`) et un nom d'hôte. Par exemple, `rabbit@ip-10-0-0-230.us-west-2.compute.internal` est un nom de nœud avec le préfixe `rabbit` et le nom d'hôte

Dimension	Description
	`ip-10-0-0-230.us-west-2.compute.internal` .
`Broker`	Nom de l'agent.

Mesures de la file d'attente RabbitMQ

Mesure	Unité	Description
`ConsumerCount`	Nombre	Nombre de consommateurs abonnés à la file d'attente.
`MessageReadyCount`	Nombre	Nombre de messages actuellement disponibles à livrer.
`MessageUnacknowledgedCount`	Nombre	Nombre de messages pour lesquels le serveur attend un accusé de réception.
`MessageCount`	Nombre	Nombre total de `MessageReadyCount` et de `MessageUnacknowledgedCount` (également appelée profondeur de file d'attente).

Dimensions pour les mesures de la file d'attente RabbitMQ

> **Note**
>
> Amazon MQ for RabbitMQ ne peut pas publier de mesures pour les hôtes virtuels et les files d'attente avec des noms contenant des espaces vides, des tabulations ou d'autres caractères non ASCII.

Pour plus d'informations sur les noms de dimension, consultez Dimension dans le manuel Amazon CloudWatch API Reference.

Dimension	Description
`Queue`	Nom de la file d'attente.
`VirtualHost`	Nom de l'hôte virtuel.
`Broker`	Nom de l'agent.

Métriques du réseau RabbitMQ

Métrique	Unité	Description
`NetworkOut`	Octets	Nombre d'octets envoyés par l'instance sur toutes les interfaces réseau. Cette métrique identifie le volume du trafic réseau sortant d'une seule instance. Le nombre mentionné correspond au nombre d'octets envoyés pendant la période. Si vous utilisez une surveillance de base (cinq minutes) et que la statistique est Somme, vous pouvez diviser ce nombre par 300 pour trouver le nombre d'octets/seconde. Si vous utilisez une surveillance détaillée (une minute) et que la statistique est Somme, divisez-la par 60. Vous pouvez également utiliser la fonction mathématique CloudWatch métrique `DIFF_TIME` pour déterminer les octets par seconde. Par exemple, si vous avez représent é graphiquement NetworkOut CloudWatch comme`m1`, la formule mathématique de la métrique `m1/(DIFF_TIME(m1))` renvoie la métrique en octets/seconde. Pour plus d'informations sur les fonctions mathématiques métriques `DIFF_TIME` et les autres, consultez la section Utilisation des mathématiques métriques.

Métrique	Unité	Description
		Statistiques significatives : somme, moyenne, minimum, maximum
`NetworkIn`	Octets	Nombre d'octets reçus par l'instance sur toutes les interfaces réseau. Cette métrique identifie le volume du trafic réseau entrant d'une seule instance. Le nombre mentionné correspond au nombre d'octets reçus pendant la période. Si vous utilisez une surveillance de base (cinq minutes) et que la statistique est Somme, vous pouvez diviser ce nombre par 300 pour trouver le nombre d'octets/seconde. Si vous utilisez une surveillance détaillée (une minute) et que la statistique est Somme, divisez-la par 60. Vous pouvez également utiliser la fonction mathématique CloudWatch métrique `DIFF_TIME` pour déterminer les octets par seconde. Par exemple, si vous avez représenté graphiquement NetworkIn CloudWatch comme `m1`, la formule mathématique de la métrique `m1/(DIFF_TIME(m1))` renvoie la métrique en octets/seconde. Pour plus d'informations sur les fonctions mathématiques métriques `DIFF_TIME` et les autres, consultez la section Utilisation des mathématiques métriques. Statistiques significatives : somme, moyenne, minimum, maximum

Dimensions pour les courtiers RabbitMQ

Dimension	Description
`BrokerId`	Identifiant du courtier

Configuration des journaux Amazon MQ pour RabbitMQ

Lorsque vous activez la CloudWatch journalisation pour vos courtiers RabbitMQ, Amazon MQ utilise un rôle lié au service pour publier des journaux généraux. CloudWatch Si aucun rôle lié au service Amazon MQ n'existe lors de la création d'un agent pour la première fois, Amazon MQ en crée automatiquement un. Tous les courtiers RabbitMQ suivants utiliseront le même rôle lié au service pour publier les journaux. CloudWatch

Pour plus d'informations sur les rôles liés à un service, consultez la section Utilisation des rôles liés à un service dans le Guide de l'utilisateur.Gestion des identités et des accès AWS Pour de plus amples informations sur la façon dont Amazon MQ utilise les rôles liés au service, consultez the section called "Utilisation des rôles liés à un service".

Journalisation des appels d'API Amazon MQ à l'aide de AWS CloudTrail

Amazon MQ est intégré à AWS CloudTrail un service qui fournit un enregistrement des appels Amazon MQ effectués par un utilisateur, un rôle ou un service. AWS CloudTrail capture les appels d'API liés aux courtiers et aux configurations Amazon MQ sous forme d'événements, y compris les appels depuis la console Amazon MQ et les appels de code depuis Amazon MQ. APIs Pour plus d'informations CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

> **Note**
>
> CloudTrail n'enregistre pas les appels d'API liés aux opérations ActiveMQ (par exemple, l'envoi et la réception de messages) ou à la console Web ActiveMQ. Pour enregistrer les informations relatives aux opérations ActiveMQ, vous pouvez configurer Amazon MQ pour publier des journaux généraux et d'audit sur Amazon Logs. CloudWatch

À l'aide des informations CloudTrail collectées, vous pouvez identifier une demande spécifique adressée à une API Amazon MQ, l'adresse IP du demandeur, son identité, la date et l'heure de la demande, etc. Si vous configurez un suivi, vous pouvez activer la diffusion continue des CloudTrail événements vers un compartiment Amazon S3. Si vous ne configurez pas de suivi, vous pouvez consulter les événements les plus récents dans l'historique des événements de la CloudTrail console. Pour plus d'informations, consultez Présentation de la création d'un journal de suivi dans le Guide de l'utilisateur AWS CloudTrail.

Informations sur Amazon MQ dans CloudTrail

Lorsque vous créez votre AWS compte, CloudTrail est activé. Lorsqu'une activité d'événement Amazon MQ prise en charge se produit, elle est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez afficher, rechercher et télécharger les événements récents pour votre compte AWS . Pour plus d'informations, consultez la section Affichage des événements avec l'historique des CloudTrail événements dans le guide de AWS CloudTrail l'utilisateur.

Un suivi permet CloudTrail de transférer des fichiers journaux vers un compartiment Amazon S3. Vous pouvez créer un parcours pour conserver une trace continue des événements sur votre AWS compte. Par défaut, lorsque vous créez un parcours à l'aide du AWS Management Console, le parcours s'applique à toutes les AWS régions. Le journal enregistre les événements de toutes les AWS régions et envoie les fichiers journaux au compartiment Amazon S3 spécifié. Vous pouvez également configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d’informations, consultez les rubriques suivantes dans le AWS CloudTrail Guide de l’utilisateur :

- CloudTrail Services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux provenant de plusieurs régions
- Réception de fichiers CloudTrail journaux provenant de plusieurs comptes

Amazon MQ prend en charge l'enregistrement des paramètres de demande et des réponses aux éléments suivants APIs sous forme d'événements dans des fichiers CloudTrail journaux :

- `CreateConfiguration`
- `DeleteBroker`
- `DeleteUser`
- `RebootBroker`
- `UpdateBroker`

> **Note**
>
> RebootBroker les fichiers journaux sont enregistrés lorsque vous redémarrez le broker. Pendant la fenêtre de maintenance, le service redémarre automatiquement et les fichiers RebootBroker journaux ne sont pas enregistrés.

> **Important**
>
> Pour les GET méthodes suivantes APIs, les paramètres de demande sont enregistrés, mais les réponses sont expurgées :
>
> - DescribeBroker
> - DescribeConfiguration
> - DescribeConfigurationRevision
> - DescribeUser
> - ListBrokers
> - ListConfigurationRevisions
> - ListConfigurations
> - ListUsers
>
> Dans les cas suivants APIs, les paramètres de `password` requête `data` et sont masqués par des astérisques () `***` :
>
> - CreateBroker (POST)
> - CreateUser (POST)
> - UpdateConfiguration (PUT)
> - UpdateUser (PUT)

Chaque événement ou entrée de journal contient des informations sur le demandeur. Cette information permet de déterminer les éléments suivants :

- La demande a-t-elle été effectuée avec les informations d'identification racine ou de l'utilisateur ?

- La demande a-t-elle été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré ?
- La demande a-t-elle été faite par un autre AWS service ?

Pour plus d'informations, consultez la section CloudTrailUserIdentity Element dans le guide de l'AWS CloudTrail utilisateur.

Exemple d'entrée de fichier journal Amazon MQ

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux au compartiment Amazon S3 spécifié. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal.

Un événement représente une demande individuelle provenant de n'importe quelle source et comprend des informations sur la demande à une API Amazon MQ, l'adresse IP du demandeur, l'identité du demandeur, la date et l'heure de l'action, etc.

L'exemple suivant montre une entrée de CloudTrail journal pour un appel d'`CreateBroker`API.

> **Note**
>
> Comme les fichiers CloudTrail journaux ne constituent pas une pile ordonnée de données publiques APIs, ils ne répertorient pas les informations dans un ordre spécifique.


```
{
    "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"
}
```


Configuration des journaux Amazon MQ pour ActiveMQ

Pour autoriser Amazon MQ à publier des journaux dans Logs, vous devez ajouter une autorisation à votre utilisateur Amazon MQ et configurer une politique basée sur les ressources pour Amazon MQ avant de créer ou de redémarrer le courtier. CloudWatch

> **Note**
>
> Lorsque vous activez les journaux et publiez des messages depuis la console Web ActiveMQ, le contenu du message est envoyé et affiché dans CloudWatch les journaux.

Ce qui suit décrit les étapes à suivre pour configurer les CloudWatch journaux de vos courtiers ActiveMQ.

Rubriques

- Comprendre la structure de la journalisation dans CloudWatch Logs
- Ajouter l'autorisation CreateLogGroup à l'utilisateur Amazon MQ
- Configurer une politique basée sur les ressources pour Amazon MQ
- Prévention du cas de figure de l'adjoint désorienté entre services

Comprendre la structure de la journalisation dans CloudWatch Logs

Vous pouvez activer la journalisation générale et la journalisation des audits lorsque vous configurez les paramètres avancés du courtier, lorsque vous créez un courtier ou lorsque vous modifiez un courtier.

La journalisation générale active le niveau de `INFO` journalisation par défaut (la `DEBUG` journalisation n'est pas prise en charge) et publie `activemq.log` dans un groupe de journaux de votre CloudWatch compte. Le groupe de journaux a un format similaire à ce qui suit :

```
/aws/amazonmq/broker/b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9/general
```

La journalisation des audits permet de consigner les actions de gestion effectuées à l'aide de JMX ou de la console Web ActiveMQ et de les `audit.log` publier dans un groupe de journaux de votre compte. CloudWatch Le groupe de journaux a un format similaire à ce qui suit :

```
/aws/amazonmq/broker/b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9/audit
```

Selon le type d'agent, à savoir un agent à instance unique ou un agent actif/en veille, Amazon MQ crée un ou deux flux de journaux dans chaque groupe de journaux. Les flux de journaux ont un format similaire à ce qui suit.

```
activemq-b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.log
                activemq-b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-2.log
```

Les suffixes `-1` et `-2` indiquent des instances d'agent individuelles. Pour plus d'informations, consultez la section Travailler avec des groupes de journaux et des flux de CloudWatch journaux dans le guide de l'utilisateur Amazon Logs.

Ajouter l'autorisation **`CreateLogGroup`** à l'utilisateur Amazon MQ

Pour autoriser Amazon MQ à créer un groupe de CloudWatch journaux Logs, vous devez vous assurer que l'utilisateur qui crée ou redémarre le broker dispose des autorisations nécessaires. `logs:CreateLogGroup`

> **Important**
>
> Si vous n'ajoutez pas l'autorisation `CreateLogGroup` à l'utilisateur Amazon MQ avant que l'utilisateur crée ou redémarre l'agent, Amazon MQ ne crée pas le groupe de journaux.

L'exemple suivant de politique axée sur IAM octroie l'autorisation pour `logs:CreateLogGroup`pour les utilisateurs auxquels cette politique est attachée.

JSON

```
{
                    "Version":"2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": "logs:CreateLogGroup",
                            "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*"
                        }
                    ]
                    }
```

> **Note**
>
> Ici, le terme « utilisateur » fait référence aux utilisateurs et non pas aux utilisateurs Amazon MQ, qui sont créés quand un nouvel agent est configuré. Pour plus d'informations sur la configuration des utilisateurs et la configuration de politiques IAM, reportez-vous à la section Présentation de la gestion des identités du Guide de l'utilisateur IAM.

Pour plus d'informations, consultez `CreateLogGroup` le manuel Amazon CloudWatch Logs API Reference.

Configurer une politique basée sur les ressources pour Amazon MQ

> **Important**
>
> Si vous ne configurez pas de politique basée sur les ressources pour Amazon MQ, le courtier ne peut pas publier les journaux dans Logs. CloudWatch

Pour autoriser Amazon MQ à publier des journaux dans votre groupe de journaux de CloudWatch journaux, configurez une politique basée sur les ressources afin de permettre à Amazon MQ d'accéder aux actions d'API de journaux suivantes : CloudWatch

- `CreateLogStream`— Crée un flux de CloudWatch journaux pour le groupe de journaux spécifié.

- PutLogEvents— Fournit des événements au flux de journal CloudWatch des journaux spécifié.

La politique basée sur les ressources suivante accorde des autorisations pour `logs:CreateLogStream` et `logs:PutLogEvents` pour. 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/*"
                            }
                        ]
                    }
```

Cette politique basée sur les ressources doit être configurée à l'aide AWS CLI de la commande suivante. Dans l'exemple, remplacez *us-east-1* avec vos propres informations.

```
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**
>
> Comme cet exemple utilise le `/aws/amazonmq/` préfixe, vous ne devez configurer la politique basée sur les ressources qu'une seule fois par AWS compte et par région.

Prévention du cas de figure de l’adjoint désorienté entre services

Le problème de député confus est un problème de sécurité dans lequel une entité qui n’est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. En AWS, l'usurpation d'identité interservices peut entraîner un problème de confusion chez les adjoints. L’usurpation d’identité entre services peut se produire lorsqu’un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d’un autre client auxquelles on ne serait pas autorisé à accéder autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services auprès des principaux fournisseurs de services qui ont obtenu l'accès aux ressources de votre compte.

Nous vous recommandons d'utiliser les clés de contexte de condition globale `aws:SourceAccount` et `aws:SourceArn` et les clés contextuelles de votre politique basée sur les ressources Amazon MQ afin de limiter l'accès aux CloudWatch journaux à un ou plusieurs courtiers spécifiés.

> **Note**
>
> Si vous utilisez les deux clés de contexte de condition globale, la valeur `aws:SourceAccount` et le compte de la valeur `aws:SourceArn` doit utiliser le même ID de compte lorsqu’il est utilisé dans la même déclaration de stratégie.

L'exemple suivant illustre une politique basée sur les ressources qui limite l'accès aux CloudWatch journaux à un seul courtier 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"
                                }
                            }
                            }
                        ]
                        }
```

Vous pouvez également configurer votre politique basée sur les ressources pour limiter l'accès aux CloudWatch journaux à tous les courtiers d'un compte, comme indiqué ci-dessous.

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"
                    }
                    }
                }
                ]
            }
```

Pour plus d'informations sur le problème de sécurité de l'adjoint confus, consultez Le problème de l'adjoint confus dans le Guide de l'utilisateur.

Résolution des problèmes de configuration des CloudWatch journaux avec Amazon MQ

Dans certains cas, CloudWatch les journaux ne se comportent pas toujours comme prévu. Cette section fournit une présentation des problèmes courants et des solutions permettant de les résoudre ou de les contourner.

Les groupes de journaux n'apparaissent pas dans CloudWatch

Ajoutez l'autorisation `CreateLogGroup` à votre utilisateur Amazon MQ et redémarrez l'agent. Cela permet à Amazon MQ de créer le groupe de journaux.

Les flux de journaux n'apparaissent pas dans les groupes de CloudWatch journaux

Configurez une politique basée sur les ressources pour Amazon MQ. Cela permet à votre agent de publier ses journaux.

Quotas dans Amazon MQ

Cette rubrique répertorie les limites au sein d'Amazon MQ. La plupart des limites suivantes peuvent être modifiées pour des AWS comptes spécifiques. Pour demander l'augmentation d'une limite, consultez Quotas de service AWS dans le Référence générale d'Amazon Web Services. Les limites actualisées ne seront pas visibles même après l'application de l'augmentation de limite. Pour plus d'informations sur l'affichage des limites de connexion actuelles sur Amazon CloudWatch, consultez la section Surveillance des courtiers Amazon MQ à l'aide d'Amazon. CloudWatch

Rubriques

Agents

Le tableau suivant répertorie les quotas relatifs aux agents Amazon MQ.

Limite	Description
Nom de l'agent	• Doit être unique dans votre AWS compte. • Doit comprendre entre 1 et 50 caractères. • Doit uniquement comprendre des caractère s spécifiés dans l'ensemble de caractères ASCII imprimables. • Peut contenir uniquement des caractères alphanumériques, des tirets, des points, des traits de soulignement et des tildes (- . _ ~).

Limite	Description
Nombre d'agents par région	50
Connexions au niveau filaire par protocole pour un agent plus petit	**Important** Ne s'applique pas aux agents RabbitMQ. 300 pour agents de type d'instance `mq.*.micro` .
Connexions au niveau filaire par protocole pour un agent plus important	**Important** Ne s'applique pas aux agents RabbitMQ. 2 000 pour agents de type d'instance `mq.*.*large` .
Groupes de sécurité par agent	5
Destinations ActiveMQ (files d'attente et sujets) surveillées dans CloudWatch	CloudWatch ne surveille que les 1 000 premières destinations.
Destinations RabbitMQ (files d'attente) surveillées dans CloudWatch	CloudWatch surveille uniquement les 500 premières destinations, classées par nombre de consommateurs.
Balises par agent	50

Configurations

Le tableau suivant répertorie les quotas relatifs aux configurations Amazon MQ.

Limite	Description
Nom de la configuration	• Doit comprendre entre 1 et 150 caractères. • Doit uniquement comprendre des caractères spécifiés dans l'ensemble de caractères ASCII imprimables. • Peut contenir uniquement des caractères alphanumériques, des tirets, des points, des traits de soulignement et des tildes (- . _ ~).
Révisions par configuration	300

Users

Le tableau suivant répertorie les quotas relatifs aux utilisateurs d'agent Amazon MQ ActiveMQ.

Limite	Description
Nom d’utilisateur	• Doit comprendre entre 1 et 100 caractères. • Doit uniquement comprendre des caractères spécifiés dans l'ensemble de caractères ASCII imprimables. • Peut contenir uniquement des caractères alphanumériques, des tirets, des points, des traits de soulignement et des tildes (- . _ ~). • Ne doit pas comporter de virgule (,).
Mot de passe	• Doit comprendre entre 12 et 250 caractères.

Limite	Description
	• Doit uniquement comprendre des caractères spécifiés dans l'ensemble de caractères ASCII imprimables. • Doit comporter au moins 4 caractères uniques. • Ne doit pas comporter de virgule (`,`).
Utilisateurs par agent (authentification simple)	250
Groupes par utilisateur (authentification simple)	20

Stockage des données

Le tableau suivant répertorie les quotas relatifs au stockage de données Amazon MQ.

Limite	Description
Capacité de stockage par agent plus petit	20 Go pour type d'instance `mq.*.micro` . Pour plus d'informations sur les types d'instance Amazon MQ, consultez Broker instance types.
Capacité de stockage par agent plus grand	200 Go pour les agents de type d'instance `mq.m5.*`. Pour plus d'informations sur les types d'instance Amazon MQ, consultez Broker instance types.
Limite d'utilisation du planificateur de tâches par agent basé sur Amazon EBS	**Important** Ne s'applique pas aux agents RabbitMQ.

Limite	Description
	50 Go Pour plus d'informations sur l'utilisation du planificateur de tâches, consultez `JobSchedulerUsage` dans la documentation sur l'API Apache ActiveMQ.
Capacité de stockage temporaire par agent plus petit.	**Important** Ne s'applique pas aux agents RabbitMQ. 5 Go pour les agents de type d'instance `mq.*.micro` .
Capacité de stockage temporaire par agent plus grand.	**Important** Ne s'applique pas aux agents RabbitMQ. 50 Go pour les agents de type d'instance `mq.m5.*`.

Restriction d'API

Les quotas de limitation suivants sont agrégés par AWS compte, sur l'ensemble des Amazon APIs MQ afin de maintenir la bande passante du service. Pour plus d'informations sur Amazon MQ APIs, consultez le manuel Amazon MQ REST API Reference.

> **Important**
>
> Ces quotas ne s'appliquent pas à Amazon MQ pour ActiveMQ ni à Amazon MQ pour la messagerie aux courtiers RabbitMQ. APIs Par exemple, Amazon MQ ne limite pas l'envoi ou la réception de messages.

Limite de rafale d'API	Limites de débit d'API
100	15

Résolutions des problèmes liés à Amazon MQ

Cette section décrit les problèmes courants que vous pouvez rencontrer lors de l'utilisation d'agents Amazon MQ et les étapes que vous pouvez suivre pour les résoudre. Pour un dépannage général, voirthe section called “Résolution des problèmes : Amazon MQ général”. Pour résoudre les problèmes liés à la version spécifique de votre moteur, consultez les sections suivantes.

Résolution des problèmes liés à ActiveMQ sur Amazon MQ

Rubrique de dépannage	Description
Résolution de problème généraux	Utilisez les informations de cette section pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontre r lorsque vous travaillez avec ActiveMQ sur Amazon MQ Brokers.
BROKER_ENI_SUPPRIMÉ	ActiveMQ sur Amazon MQ déclenche `BROKER_ENI_DELETED` une alarme lorsque vous supprimez l'Elastic Network Interface (ENI) d'un courtier.
BROKER OOM	ActiveMQ sur Amazon MQ déclenche une alarme BROKER_OOM lorsque le broker effectue une boucle de redémarrage en raison d'une capacité de mémoire insuffisante

Résolution des problèmes liés à RabbitMQ sur Amazon MQ

Rubrique de dépannage	Description
Résolution de problème généraux	Diagnostiquez les problèmes courants que vous pourriez rencontrer lorsque vous

Rubrique de dépannage	Description
	travaillez avec des courtiers RabbitMQ.
RABBITMQ_MEMORY_ALARM	RabbitMQ déclenchera une alarme de mémoire élevée lorsque l'utilisation de la mémoire par le courtier, identifiée par CloudWatch métrique`RabbitMQMemUsed` , dépasse la limite de mémoire, identifiée par. `RabbitMQMemLimit`
RABBITMQ_KMS_KEY INVALIDE	RabbitMQ sur Amazon MQ génère un code d'action critique INVALID_KMS_KEY requis lorsqu'un courtier créé avec une solution gérée par le client AWS KMS key(CMK) détecte que la clé (KMS) est désactivée. AWS Key Management Service
RABBITMQ_INVALID_ASSUMEROLE	RabbitMQ sur Amazon MQ génère un code d'action critique INVALID_ASSUMEROLE requis lorsque l'ARN du rôle IAM spécifié dans ne peut pas être assumé par Amazon MQ. `aws.arns.assume_role_arn`

Rubrique de dépannage	Description
RABBITMQ_INVALID_ARN_LDAP	RabbitMQ sur Amazon MQ génère un code d'action critique INVALID_ARN_LDAP requis lorsque l'ARN du mot de passe du compte de service LDAP n'est pas valide ou n'est pas accessible.
RABBITMQ_INVALID_ARN_HTTP	RabbitMQ sur Amazon MQ génère un code d'action critique INVALID_ARN_HTTP requis lorsqu'un ou plusieurs certificats SSL ou fichier clé pour HTTP auth_backend ne sont ARNs pas valides ou inaccessibles.
RABBITMQ_INVALID_ARN_SSL	RabbitMQ sur Amazon MQ génère un code d'action critique INVALID_ARN_SSL requis lorsqu'un ou plusieurs certificats CA Truststore pour EXTERNAL auth_mechanism ne sont pas valides ou ARNs inaccessibles.
RABBITMQ_ARN_INVALIDE	RabbitMQ sur Amazon MQ génère un code d'action critique INVALID_ARN requis lorsqu'un ou plusieurs ARNs éléments de la configuration du broker ne sont pas valides ou inaccessibles.

Rubrique de dépannage	Description
RABBITMQ_DISK_ALARM	L'alarme de limite de disque indique que le volume de disque utilisé par un nœud RabbitMQ a diminué en raison d'un nombre élevé de messages non consommés lors de l'ajout de nouveaux messages.

Résolution des problèmes : Amazon MQ général

Utilisez les informations de cette section pour identifier les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec des agents Amazon MQ, tels que les problèmes de connexion à votre agent et les redémarrages des agents.

Table des matières

Je ne parviens pas à me connecter à la console web ou aux points de terminaison de mon agent.

Si vous rencontrez des problèmes de connexion à votre agent à l'aide de la console web ou des points de terminaison filaires, nous vous recommandons les étapes suivantes.

1. Vérifiez si vous tentez de vous connecter à votre agent à partir de derrière un pare-feu. Vous devrez peut-être configurer le pare-feu pour autoriser l'accès à votre agent.
2. Vérifiez si vous essayez de vous connecter à votre agent à l'aide d'un point de terminaison FIPS. Amazon MQ ne prend en charge les points de terminaison FIPS que lors des opérations d'API, et non pour les connexions au niveau filaire à l'instance d'agent elle-même.

3. Vérifiez si l'option Public Accessibility (Accessibilité publique) pour votre agent est définie sur Yes (Oui). Si cette valeur est définie sur No (Non), vérifiez les règles de Liste de contrôle d'accès (ACL) du réseau de votre sous-réseau. Si vous avez créé un réseau personnalisé ACLs, vous devrez peut-être modifier les règles ACL du réseau pour permettre l'accès à votre courtier. Pour plus d'informations sur le réseau Amazon VPC, consultez la section Activation de l'accès à Internet dans le guide de l'utilisateur Amazon VPC
4. Vérifiez les règles du groupe de sécurité de votre agent. Assurez-vous que vous autorisez les connexions aux ports suivants :

> **Note**
>
> Les ports suivants sont regroupés en fonction du type de moteur, car ActiveMQ sur Amazon MQ et RabbitMQ sur Amazon MQ utilisent des ports différents pour les connexions.

ActiveMQ sur Amazon MQ

- Console Web – Port 8162
- OpenWire — Port 61617
- AMQP – Port 5671
- STOMP – Port 61614
- MQTT – Port 8883
- WSS – Port 61619

RabbitMQ sur Amazon MQ

- Console web et API de gestion – Port 443 et 15671
- AMQP – Port 5671

5. Exécutez les tests de connectivité réseau suivants pour votre type de moteur d'agent.

> **Note**
>
> Pour les agents sans accès public, exécutez les tests à partir d'une instance Amazon EC2 (Amazon EC2) au sein du même Amazon VPC que votre agent Amazon MQ (Amazon MQ) et évaluez les réponses.

ActiveMQ on Amazon MQ

Pour tester la connectivité réseau de votre courtier ActiveMQ sur Amazon MQ

1. Ouvrez une fenêtre de terminal ou de ligne de commande.
2. Exécutez la commande `nslookup` suivante pour interroger votre enregistrement DNS de l'agent. Pour des déploiements actifs/en veille, testez à la fois les points de terminaison actifs et en veille. Les active/standby points de terminaison sont identifiés par un suffixe `-1` ou `-2` ajoutés à l'identifiant unique du courtier. Remplacez le point de terminaison par vos informations.

```
$ nslookup b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
```

Si la requête aboutit, vous obtenez un résultat similaire à ce qui suit.

```
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'adresse IP résolue doit correspondre aux adresses IP fournies dans la console Amazon MQ. Cela indique que le nom de domaine est correctement résolu sur le serveur DNS et que vous pouvez passer à l'étape suivante.

3. Exécutez la commande `telnet` suivante pour tester le chemin réseau de votre agent. Remplacez le point de terminaison par vos informations. *`port`*Remplacez-le par le numéro 8162 de port de la console Web ou par d'autres ports au niveau du fil pour tester des protocoles supplémentaires selon les besoins.

> **Note**
>
> Pour les active/standby déploiements, vous recevrez un message d'`Connect failed`erreur si vous exécutez `telnet` avec le point de terminaison de secours. Ceci est attendu, car l'instance en veille elle-même est en cours d'exécution, mais le processus ActiveMQ n'est pas en cours d'exécution et n'a pas accès au volume

de stockage Amazon EFS de l'agent. Exécutez la commande pour les points de terminaison `-1` et `-2` pour vous assurer de tester à la fois les instances actives et les instances en veille.

```
$ telnet b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com port
```

Pour l'instance active, vous obtenez une sortie similaire à ce qui suit.

```
Connected to b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com.
Escape character is '^]'.
```

4. Effectuez l'une des actions suivantes :

   - Si la commande `telnet` réussit, vérifiez la mesure `EstablishedConnectionsCount` et confirmez que l'agent n'a pas atteint la limite de connexion au niveau filaire maximum. Vous pouvez également confirmer si la limite a été atteinte en consultant les journaux `General` d'agent. Si cette mesure est supérieure à zéro, alors au moins un client est actuellement connecté à l'agent. Si la mesure affiche zéro connexion, exécutez le test de chemin `telnet` de nouveau et attendez au moins une minute avant de déconnecter, car les mesures de l'agent sont publiées toutes les minutes.
   - Si la commande `telnet` échoue, vérifiez l'état de l'interface réseau Elastic de votre agent, et confirmez que l'état est `in-use`. Créez un journal de flux Amazon VPC pour l'interface réseau de chaque instance et passez en revue les journaux de flux générés. Recherchez les adresses IP de l'agent lorsque vous avez exécuté la commande `telnet` et confirmez que les paquets de connexion sont `ACCEPTED`, y compris un paquet de retour. Pour plus d'informations et pour voir un exemple de journal de flux, reportez-vous à Exemples d'enregistrements de journaux de flux dans le Guide du développeur Amazon VPC.

5. Exécutez la commande `curl` suivante pour vérifier la connectivité à la console web d'administration ActiveMQ.

```
$ curl https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com:8162/index.html
```

Si la commande aboutit, la sortie doit être un document HTML similaire à ce qui suit.

```
<!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

Pour tester la connectivité réseau de votre RabbitMQ sur Amazon MQ Broker

1. Ouvrez une fenêtre de terminal ou de ligne de commande.
2. Exécutez la commande `nslookup` suivante pour interroger l'enregistrement DNS de votre agent. Remplacez le point de terminaison par vos informations.

   ```
   $ nslookup b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
   ```

   Si la requête aboutit, vous obtenez un résultat similaire à ce qui suit.

   ```
   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. Exécutez la commande `telnet` suivante pour tester le chemin réseau de votre agent. Remplacez le point de terminaison par vos informations. Vous pouvez le *`port`* remplacer par un port `443` pour la console Web et `5671` pour tester la connexion AMQP au niveau du fil.

```
$ telnet b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com port
```

Si la commande aboutit, vous obtenez un résultat similaire à ce qui suit.

```
Connected to b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com.
Escape character is '^]'.
```

> **Note**
>
> La connexion telnet se ferme automatiquement après quelques secondes.

4. Effectuez l'une des actions suivantes :

   - Si la commande `telnet` réussit, vérifiez la mesure `ConnectionCount` et confirmez que l'agent n'a pas atteint la valeur définie dans la politique par défaut `max-connections`. Vous pouvez également confirmer si la limite a été atteinte en consultant le groupe de journaux `Connection.log` de l'agent. Si cette mesure est supérieure à zéro, il y a au moins un client actuellement connecté à l'agent. Si la mesure affiche zéro connexion, exécutez alors le test de chemin `telnet` à nouveau. Vous devrez peut-être répéter ce processus si la connexion se ferme avant que votre courtier n'ait publié de nouvelles métriques de connexion sur CloudWatch. Les mesures sont publiées toutes les minutes.
   - Pour les agents qui n'ont pas d'accès public, si la commande `telnet` échoue, vérifiez l'état de l'interface réseau Elastic de votre agent et confirmez que l'état est `in-use`. Créez un journal de flux Amazon VPC pour chaque interface réseau et examinez les journaux de flux générés. Recherchez les adresses IP privées de l'agent lorsque la commande `telnet` a été appelée et confirmez que les paquets de connexion sont `ACCEPTED`, y compris un paquet de retour. Pour plus d'informations et pour voir un exemple de journal de flux, reportez-vous à Exemples d'enregistrements de journaux de flux dans le Guide du développeur Amazon VPC.

> **Note**
>
> Cette étape ne s'applique pas aux courtiers RabbitMQ sur Amazon MQ accessibles au public.

5. Exécutez la commande `curl` suivante pour vérifier la connectivité à la console web d'administration RabbitMQ.

```
$ curl https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com:443/index.html
```

Si la commande aboutit, la sortie doit être un document HTML similaire à ce qui suit.

```
<!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>
        ...
```

Mon agent est en cours d'exécution et je peux vérifier la connectivité via **`telnet`**, mais mes clients ne peuvent pas se connecter et renvoient des exceptions SSL.

Le certificat de point de terminaison de votre agent a peut-être été mis à jour pendant la fenêtre de maintenance de celui-ci. Les certificats d'agent Amazon MQ font l'objet d'une rotation périodique pour garantir la disponibilité et la sécurité continues des agents.

Nous recommandons d'utiliser l'autorité de certification (CA) racine d'Amazon dans les services Amazon Trust pour vous authentifier dans le magasin de clés de confiance de vos clients. Tous les certificats des agents Amazon MQ portent la signature de cette autorité de certification racine. Le CA racine Amazon vous permet d'éviter d'avoir à télécharger le nouveau certificat d'agent Amazon MQ chaque fois qu'il est mis à jour.

J’ai créé un agent mais la création de l’agent a échoué.

Si l’état de votre agent est `CREATION_FAILED`, procédez comme suit.

- Vérifiez vos autorisations IAM. Pour créer un courtier, vous devez soit utiliser la politique IAM AWS gérée, `AmazonMQFullAccess` soit disposer du bon ensemble d'autorisations Amazon EC2 dans votre politique IAM personnalisée. Pour en savoir plus sur les autorisations Amazon EC2 dont vous avez besoin, reportez-vous à Autorisations IAM requises pour créer un agent Amazon MQ.
- Vérifiez si le sous-réseau que vous choisissez pour votre agent se trouve dans un Amazon Virtual Private Cloud (VPC) partagé. Pour créer un agent Amazon MQ dans un Amazon VPC partagé, vous devez le créer dans le compte qui possède l’Amazon VPC.

Mon agent a redémarré et je ne sais pas pourquoi.

Si votre agent a redémarré automatiquement, cela peut être dû à l’une des raisons suivantes.

- Votre agent a peut-être redémarré en raison d’une fenêtre de maintenance hebdomadaire planifiée. Périodiquement, Amazon MQ effectue la maintenance du matériel, du système d’exploitation ou du logiciel moteur d’un agent de messages. La durée de la maintenance varie, mais peut durer jusqu’à deux heures, selon les opérations planifiées pour votre agent de messages. Les agents peuvent redémarrer à tout moment pendant la période de maintenance de deux heures. Pour plus d'informations sur les fenêtres de maintenance des courtiers, consultezthe section called “Planification de la maintenance des courtiers”.
- Votre type d’instance d’agent peut ne pas convenir à la charge de travail de votre application. Par exemple, l’exécution d’une charge de travail de production sur un `mq.t3.micro` peut entraîner l’agent à être à court de ressources. Une utilisation élevée du processeur ou une utilisation élevée de la mémoire de l’agent peuvent entraîner un redémarrage inattendu d’un agent. Pour connaître la quantité de processeur et de mémoire utilisée par votre courtier, utilisez les CloudWatch mesures suivantes pour votre type de moteur.
  - ActiveMQ sur Amazon MQ : `CpuUtilization` vérifiez le pourcentage d'unités de calcul Amazon EC2 allouées que le courtier utilise actuellement. Vérifiez `HeapUsage` pour le pourcentage de la limite de mémoire JVM ActiveMQ actuellement utilisée par l’agent.
  - RabbitMQ sur Amazon MQ — Vérifiez le pourcentage d'unités `SystemCpuUtilization` de calcul Amazon EC2 allouées que le courtier utilise actuellement. Vérifiez `RabbitMQMemUsed` pour le volume de RAM utilisé en octets, et divisez par `RabbitMQMemLimit` pour le pourcentage de mémoire utilisé par le nœud RabbitMQ.

Pour plus d'informations sur les types d'instances de courtier et sur la manière de choisir le type d'instance adapté à votre charge de travail, consultezBroker instance types.

Résolution des problèmes liés à ActiveMQ sur Amazon MQ

Utilisez les informations de cette section pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec ActiveMQ sur Amazon MQ Brokers.

Table des matières

Je ne peux pas voir les journaux généraux ou d'audit de mon courtier dans CloudWatch Logs, même si j'ai activé la journalisation.

Si vous ne parvenez pas à consulter les journaux de votre courtier dans CloudWatch Logs, procédez comme suit.

1. Vérifiez si l’utilisateur qui crée ou redémarre l’agent dispose de l’autorisation `logs:CreateLogGroup`. Si vous n'ajoutez pas l'autorisation `CreateLogGroup` à l'utilisateur avant que l'utilisateur crée ou redémarre l'agent, Amazon MQ ne crée pas le groupe de journaux.
2. Vérifiez si vous avez configuré une politique basée sur les ressources pour autoriser Amazon MQ à publier des journaux dans Logs. CloudWatch Pour autoriser Amazon MQ à publier des journaux dans votre groupe de journaux de CloudWatch journaux, configurez une politique basée sur les ressources afin de permettre à Amazon MQ d'accéder aux actions d'API de journaux suivantes : CloudWatch
   - `CreateLogStream`— Crée un flux de CloudWatch journaux pour le groupe de journaux spécifié.

- `PutLogEvents`— Fournit des événements au flux de journal CloudWatch des journaux spécifié.

Pour plus d'informations sur la configuration d'ActiveMQ sur Amazon MQ pour publier des journaux dans des journaux CloudWatch , consultez Configuration de la journalisation.

Après le redémarrage ou la fenêtre de maintenance de l'agent, je ne peux pas me connecter à mon agent même si le statut est **`RUNNING`**. Pourquoi ?

Il se peut que vous rencontriez des problèmes de connexion après le redémarrage d½un agent que vous avez initié, après la fin d'une fenêtre de maintenance planifiée, ou lors d'un événement d'échec, où l'instance de secours est activée. Dans les deux cas, les problèmes de connexion après le redémarrage d'un agent sont très probablement causés par un nombre anormalement élevé de messages persistants dans le volume de stockage Amazon EFS ou Amazon EBS de votre agent. Lors d'un redémarrage, Amazon MQ déplace les messages persistants du stockage vers la mémoire de l'agent. Pour confirmer ce diagnostic, vous pouvez surveiller les mesures suivantes CloudWatch pour votre courtier Amazon MQ pour ActiveMQ :

- **`StorePercentUsage`** — Des pourcentages élevés à 100 % ou près de 100 % peuvent amener l'agent à refuser des connexions.
- **`JournalFilesForFullRecovery`** — Indique le nombre de fichiers journaux qui seront réutilisés après un arrêt et un redémarrage incorrect. Une valeur croissante, ou constamment supérieure à un, indique des transactions non résolues qui peuvent causer des problèmes de connexion après le redémarrage.
- **`OpenTransactionCount`** — Un nombre supérieur à zéro à la suite d'un redémarrage indique que l'agent tentera de stocker les messages précédemment consommés, provoquant ainsi des problèmes de connexion.

Pour résoudre ce problème, nous vous recommandons de résoudre vos transactions XA à l'aide d'un `rollback()` ou un `commit()`. Pour plus d'informations et pour voir un exemple de code de résolution de transactions XA à l'aide de`rollback()`, consultez récupération de transactions XA.

Certains de mes clients se connectent à l'agent, tandis que d'autres ne peuvent pas se connecter.

Si le statut de votre agent est `RUNNING` et que certains clients parviennent à se connecter à l'agent avec succès, alors que d'autres n'y parviennent pas, il se peut que vous ayez atteint la limite de

connexions au niveau filaire pour l'agent. Pour vérifier que vous avez atteint la limite de connexions au niveau filaire, procédez comme suit :

- Consultez les journaux généraux de votre courtier ActiveMQ sur Amazon MQ dans Logs. CloudWatch Si la limite a été atteinte, vous verrez `Reached Maximum Connections` dans les journaux de l'agent. Pour plus d'informations sur CloudWatch Logs for ActiveMQ sur les courtiers Amazon MQ, consultez. the section called “Comprendre la structure de la journalisation dans CloudWatch Logs”

Une fois que la limite de connexions au niveau filaire est atteinte, l'agent refuse activement les connexions entrantes supplémentaires. Pour résoudre ce problème, nous vous recommandons de mettre à niveau le type d'instance de votre agent. Pour plus d'informations sur le choix du meilleur type d'instance pour votre application, consultez la section Broker instance types.

Si vous avez confirmé que le nombre de vos connexions filaires est inférieur à la limite de connexion de l'agent, le problème peut être lié au redémarrage des clients. Vérifiez dans les journaux de votre agent s'il y a des entrées nombreuses et fréquentes de `... Inactive for longer than 600000 ms - removing ...`. L'entrée du journal indique un redémarrage des clients ou des problèmes de connectivité. Cet effet est plus évident lorsque les clients se connectent à l'agent via un Network Load Balancer avec des clients qui se déconnectent et se reconnectent fréquemment à l'agent. Ceci est typiquement observé dans les clients axés sur des conteneurs.

Vérifiez vos journaux côté client pour plus de informations. L'agent nettoie les connexions TCP inactives après 600 000 ms et libère le socket de connexion.

Je vois une exception **`org.apache.jasper.JasperException: An exception occurred processing JSP page`** sur la console ActiveMQ lorsque j'effectue des opérations.

Si vous utilisez l'authentification simple et configurez `AuthorizationPlugin` pour l'autorisation des files d'attente et des rubriques, assurez-vous d'utiliser l'élément `AuthorizationEntries` dans votre fichier de configuration XML, et autorisez le groupe `activemq-webconsole` à toutes les files d'attente et les rubriques. Cela garantit que la console web ActiveMQ peut communiquer avec l'agent ActiveMQ.

L'exemple `AuthorizationEntry` suivant accorde des autorisations de lecture et d'écriture pour toutes les files d'attente et les rubriques au groupe `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>
```

De même, lorsque vous intégrez votre agent à LDAP, assurez-vous d'accorder l'autorisation pour le groupe `amazonmq-console-admins`. Pour plus d'informations sur l'intégration LDAP, consultezthe section called “Fonctionnement de l'intégration avec LDAP”.

Résolution des problèmes : RabbitMQ sur Amazon MQ

Utilisez les informations de cette section pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pourriez rencontrer lorsque vous travaillez avec RabbitMQ sur Amazon MQ Brokers.

Table des matières

Je ne peux pas voir les statistiques relatives à mes files d'attente ou à CloudWatch mes hôtes virtuels.

Si vous ne parvenez pas à consulter les statistiques relatives à vos files d'attente ou à vos hôtes virtuels CloudWatch, vérifiez si les noms de vos files d'attente ou d'hôtes virtuels contiennent des espaces, des onglets ou d'autres caractères non ASCII.

Amazon MQ ne peut pas publier de mesures pour les hôtes virtuels et les files d'attente avec des noms contenant des espaces vides, des tabulations ou d'autres caractères non ASCII.

Pour plus d'informations sur les noms des dimensions, consultez Dimension dans le manuel Amazon CloudWatch API Reference.

Comment activer les plugins dans RabbitMQ sur Amazon MQ ?

RabbitMQ sur Amazon MQ ne prend actuellement en charge que le plugin de gestion, de pelle, de fédération et d'échange de hachage cohérent de RabbitMQ, qui est activé par défaut. Pour plus d'informations sur l'utilisation des plugins pris en charge, consultez the section called “Plug-ins”.

Je ne parviens pas à modifier la configuration Amazon VPC pour l'agent.

Amazon MQ ne prend pas en charge la modification de la configuration Amazon VPC après la création de votre agent. Veuillez noter que vous devrez créer un nouveau agent avec la nouvelle configuration Amazon VPC et mettre à jour l'URL de connexion du client avec la nouvelle URL de connexion de l'agent.

Les déploiements de clusters ont suspendu mes synchronisations de files d'attente.

Lorsque vous répondez aux alarmes de mémoire élevée de RabbitMQ, vous pouvez constater que les messages sur une ou plusieurs files d'attente ne peuvent pas être consommés. Ces files d'attente peuvent être en cours de synchronisation des messages entre les nœuds, au cours desquels les files d'attente respectives deviennent indisponibles pour publication et consommation. Les synchronisations des files d'attente peuvent être suspendues en raison de l'alarme de mémoire élevée et même contribuer à l'alarme de mémoire.

Pour plus d'informations sur l'arrêt et la nouvelle tentative de synchronisation de la file d'attente en pause, veuillez consulter the section called “Résolution de la synchronisation des files d'attente mises en pause”.

Mon broker à instance unique Amazon MQ pour RabbitMQ est dans une boucle de redémarrage.

Un broker à instance unique Amazon MQ pour RabbitMQ qui déclenche une alarme d'insuffisance de mémoire risque de devenir indisponible s'il redémarre et ne dispose pas de suffisamment de mémoire pour démarrer. Cela peut amener RabbitMQ à entrer dans une boucle de redémarrage et à empêcher toute autre interaction avec l'agent jusqu'à ce que le problème soit résolu. Si votre courtier est dans

une boucle de redémarrage, vous ne serez pas en mesure d'appliquer les meilleures pratiques recommandées par Amazon MQ pour résoudre l'alarme d'insuffisance de mémoire.

Pour récupérer votre agent, nous vous recommandons de passer à un type d'instance plus grand avec plus de mémoire. Contrairement aux déploiements en cluster, vous pouvez mettre à niveau un broker à instance unique lorsqu'il reçoit une alarme de mémoire insuffisante, car aucune synchronisation de file d'attente n'est à effectuer entre les nœuds lors d'un redémarrage.

J'ai perdu l'accès à tous les comptes d'administrateur de mon courtier.

Vous pouvez récupérer l'accès à l'aide de l'authentification IAM. Activez la fédération d'identité Web sortante pour votre AWS compte, créez un rôle IAM autorisé à obtenir des jetons d'identité Web, configurez votre courtier pour qu'il accepte l'authentification IAM via la OAuth version 2.0, puis utilisez les informations d'identification IAM pour obtenir un jeton JWT et créez un nouvel utilisateur administrateur. Pour obtenir des instructions complètes, consultez the section called “Utilisation de l'authentification et de l'autorisation IAM”.

ActiveMQ sur Amazon MQ : alarme supprimée de l'interface réseau élastique

ActiveMQ sur Amazon MQ déclenche une alarme BROKER_ENI_DELETED lorsque vous supprimez l'Elastic Network Interface (ENI) d'un courtier. Lorsque vous créez un agent Amazon MQ pour la première fois, Amazon MQ alloue une interface réseau Elastic dans le Virtual Private Cloud (VPC) sous votre compte et demande un nombre d'autorisations EC2.

Vous ne devez pas modifier ou supprimer cette interface réseau. La modification ou la suppression de l'interface réseau peut entraîner une perte définitive de la connexion entre votre VPC et votre agent. Si vous souhaitez supprimer l'interface réseau, vous devez d'abord supprimer l'agent.

ActiveMQ sur Amazon MQ : alarme de manque de mémoire pour le courtier

ActiveMQ sur Amazon MQ déclenche une alarme BROKER_OOM lorsque le broker effectue une boucle de redémarrage en raison d'une capacité de mémoire insuffisante. Lorsqu'un agent est dans une boucle de redémarrage, également appelée boucle de rebond, il effectue des tentatives de restauration répétées dans un court laps de temps. Les agents qui ne peuvent pas terminer le

démarrage en raison d'une capacité insuffisante de la mémoire peuvent entrer dans une boucle de redémarrage, au cours de laquelle les interactions avec l'agent sont limitées.

Amazon MQ active les mesures pour votre agent par défaut. Vous pouvez consulter les statistiques de votre courtier en accédant à la CloudWatch console Amazon ou en utilisant l' CloudWatch API. Les mesures suivantes sont utiles lors du diagnostic de l'alarme ActiveMQ BROKER_OOM :

Métrique Amazon MQ CloudWatch	Raison de l'utilisation élevée de la mémoire	
`TotalMessageCount`	Les messages sont stockés en mémoire jusqu'à ce qu'ils soient consommés ou rejetés. Un nombre élevé de messages peut indiquer une surutilisation des ressources et peut entraîner une alarme de mémoire élevée.	
`HeapUsage`	Pourcentage de la limite de mémoire JVM ActiveMQ actuellement utilisé par l'agent. Un pourcentage supérieur indique que l'agent utilise des ressources importantes et peut entraîner une alarme OOM.	
`ConnectionCount`	Les connexions client utilisent de la mémoire, et un trop grand nombre de connexions simultanées peuvent entraîner une alarme de mémoire élevée.	
`CpuUtilization`	Pourcentage d'unités de calcul EC2 allouées actuellem ent utilisées par l'agent.	

Métrique Amazon MQ CloudWatch	Raison de l'utilisation élevée de la mémoire	
`TotalConsumerCount`	Pour chaque consommateur connecté à l'agent, un certain nombre de messages sont chargés depuis le stockage dans la mémoire avant d'être remis au consommateur. Un grand nombre de connexions grand public peuvent entraîner une utilisation élevée de la mémoire et entraîner une alarme de mémoire élevée.	

Pour éviter les boucles de redémarrage et l'apparition de l'alarme BROKER_OOM, assurez-vous que les messages sont consommés rapidement. Pour ce faire, vous pouvez choisir le type d'instance d'agent le plus efficace et nettoyer votre file d'attente de lettres mortes pour supprimer les messages non distribuables ou expirés. Pour en savoir plus sur la garantie de performances efficaces chez ActiveMQ, consultez les meilleures pratiques d'Amazon MQ.

Amazon MQ pour RabbitMQ : alarme de mémoire élevée

Amazon MQ pour RabbitMQ déclenche une alarme de mémoire élevée lorsque l'utilisation de la mémoire par le courtier, identifiée par CloudWatch métrique, dépasse la limite de mémoire`RabbitMQMemUsed`, identifiée par. `RabbitMQMemLimit`

Un courtier RabbitMQ qui a déclenché une alarme de mémoire trop élevée bloquera tous les clients qui publient des messages. Votre courtier peut entrer dans une boucle de redémarrage, connaître une interruption de la synchronisation des files d'attente ou développer d'autres problèmes qui compliquent le diagnostic et la résolution de l'alarme.

Pour diagnostiquer et résoudre une alarme d'insuffisance de mémoire, suivez d'abord toutes les meilleures pratiques pour RabbitMQ, puis effectuez les étapes suivantes.

⚠ Important

- `RabbitMQMemLimit`est défini par Amazon MQ et est spécifiquement réglé en fonction de la mémoire disponible pour chaque type d'instance hôte.
- Amazon MQ ne redémarre pas un agent confronté à une alarme de mémoire élevée et retournera une exception pour `RebootBroker` Opérations API tant que l'agent continue de déclencher l'alarme.

Étape 1 : Diagnostiquer une alarme de mémoire trop importante

Il existe deux méthodes pour diagnostiquer les alarmes de mémoire trop importante sur votre courtier Amazon MQ for RabbitMQ. Nous vous recommandons de vérifier à la fois la console Web RabbitMQ et les métriques Amazon MQ dans. CloudWatch

Diagnostiquez une alarme de mémoire insuffisante à l'aide de la console Web RabbitMQ

La console Web RabbitMQ peut générer et afficher des informations détaillées sur l'utilisation de la mémoire pour chaque nœud. Vous pouvez trouver ces informations en suivant les procédures ci-dessous :

1. Connectez-vous à la console AWS Management Console Web RabbitMQ de votre courtier et ouvrez-la.
2. Sur la console RabbitMQ, sur la Présentation, choisissez le nom d'un nœud dans la liste deNœuds.
3. Sur la page des informations du nœud, choisissez Informations sur la mémoire pour développer la section afin d'afficher les informations d'utilisation de la mémoire du nœud.

Les informations d'utilisation de la mémoire fournies par RabbitMQ dans la console Web peuvent vous aider à déterminer quelles ressources peuvent consommer trop de mémoire et contribuer à l'alarme de mémoire élevée. Pour plus d'informations sur les détails de l'utilisation de la mémoire disponibles via la console Web RabbitMQ, voir Reasoning About Memory Use sur le site Web de documentation du serveur RabbitMQ.

Diagnostiquez une alarme de mémoire trop importante à l'aide des métriques Amazon MQ

Amazon MQ active les mesures pour votre agent par défaut. Vous pouvez consulter les statistiques de votre courtier en accédant à la CloudWatch console ou en utilisant l' CloudWatch API. Les mesures suivantes sont utiles lors du diagnostic de l’alarme à mémoire élevée RabbitMQ.

Métrique Amazon MQ CloudWatch	Raison de l’utilisation élevée de la mémoire
`MessageCount`	Les messages sont stockés en mémoire jusqu’à ce qu’ils soient consommés ou rejetés. Un nombre élevé de messages peut indiquer une surutilisation des ressources et peut entraîner une alarme de mémoire élevée.
`QueueCount`	Les files d'attente sont stockées en mémoire et un nombre élevé de files d'attente peut entraîner une alarme de mémoire élevée.
`ConnectionCount`	Les connexions client utilisent de la mémoire, et un trop grand nombre de connexions simultanées peuvent entraîner une alarme de mémoire élevée.
`ChannelCount`	Comme dans le cadre des connexions, les canaux établis à l'aide de chaque connexion sont également stockés dans la mémoire des nœuds, et un nombre élevé de canaux

Métrique Amazon MQ CloudWatch	Raison de l'utilisation élevée de la mémoire	
	peut entraîner une alarme de mémoire élevée.	
`ConsumerCount`	Pour chaque consommateur connecté à l'agent, un certain nombre de messages sont chargés depuis le stockage dans la mémoire avant d'être remis au consommateur. Un grand nombre de connexions grand public peuvent entraîner une utilisation élevée de la mémoire et entraîner une alarme de mémoire élevée.	
`PublishRate`	La publication de messages utilise la mémoire de l'agent. Si la fréquence taux de publication des messages sur l'agent est trop élevé et dépasse considérablement le taux auquel l'agent envoie des messages aux consommat eurs, l'agent peut déclenche r une alarme de mémoire élevée.	

Étape 2 : Corriger et empêcher l'alarme de mémoire trop importante

> **Note**
>
> Plusieurs heures peuvent être nécessaires pour que le statut de RABBITMQ_MEMORY_ALARM s'efface une fois que vous avez effectué les actions requises.

Suivez toutes les meilleures pratiques pour RabbitMQ en tant que méthode générale de prévention. Pour chaque contributeur spécifique que vous identifiez, nous recommandons l'ensemble d'actions suivant pour traiter et prévenir les alarmes de mémoire trop importantes de RabbitMQ.

Source d'utilisation élevée de la mémoire	Recommandation d'Amazon MQ pour l'adressage	Recommandation d'Amazon MQ pour prévenir
Nombre de messages	Consommez les messages publiés dans les files d'attente, purgez les messages des files d'attente ou supprimez les files d'attente de votre courtier.	Activez les files d'attente latentes et définissez ou réduisez la limite de profondeur des files d'attente.
Nombre de files d'attente	Réduisez le nombre de files d'attente.	Définissez ou réduisez le nombre limite de files d'attente .
Nombre de connexions	Réduisez le nombre de connexions.	Définissez ou réduisez la limite du nombre de connexions.
Nombre de canaux	Réduisez le nombre de canaux.	Définissez un nombre maximal de canaux par connexion sur les applications clientes.
Nombre de consommateurs	Réduire le nombre de consommateurs connectés à l'agent.	Définissez une petite limite de prérécupération pour les consommateurs.

Source d'utilisation élevée de la mémoire	Recommandation d'Amazon MQ pour l'adressage	Recommandation d'Amazon MQ pour prévenir
Taux de publication des messages	Réduisez la fréquence à laquelle les éditeurs envoient des messages à l'agent.	Activez les confirmations de l'éditeur.
Taux de tentatives de connexion client	Réduisez la fréquence à laquelle les clients tentent de se connecter à l'agent afin de publier ou de consommer des messages, ou de configurer l'agent.	Utilisez des connexions à plus longue durée pour réduire le nombre et la fréquence des tentatives de connexion.

Une fois l'alarme de mémoire de votre courtier résolue, vous pouvez mettre à niveau votre type d'instance hôte vers une instance dotée de ressources supplémentaires. Pour plus d'informations sur la façon de mettre à jour le type d'instance de votre courtier, consultez `UpdateBrokerInput`le manuel Amazon MQ REST API Reference.

> **Note**
>
> Vous ne pouvez pas rétrograder un broker d'un type d'`mq.m5.x`instance à un type d'`mq.t3.micro`instance. Pour rétrograder, vous devez supprimer votre courtier et en créer un nouveau.

RabbitMQ sur Amazon MQ : clé non valide AWS Key Management Service

RabbitMQ sur Amazon MQ génère un code d'action critique INVALID_KMS_KEY requis lorsqu'un courtier créé avec une solution gérée par le client AWS KMS key(CMK) détecte que la clé (KMS) est désactivée. AWS Key Management Service Un agent RabbitMQ doté d'une clé CMK vérifie régulièrement que la clé KMS est activée et que l'agent dispose de toutes les autorisations nécessaires. Si RabbitMQ ne peut pas vérifier que la clé est activée, l'agent est mis en quarantaine et RabbitMQ renverra INVALID_KMS_KEY.

Sans clé KMS active, l'agent ne dispose pas des autorisations de base pour les clés KMS gérées par le client. L'agent ne peut pas effectuer d'opérations de chiffrement avec votre clé tant que vous n'aurez pas réactivé votre clé et que l'agent n'aura pas redémarré. Un agent RabbitMQ avec une clé KMS désactivée est mis en quarantaine pour éviter toute détérioration. Une fois que RabbitMQ a déterminé que la clé KMS est de nouveau active, votre agent est retiré de la quarantaine. Amazon MQ ne redémarre pas d'agent avec une clé KMS désactivée et retourne une exception pour les opérations d'API `RebootBroker` tant que l'agent continue de disposer d'une clé KMS non valide.

Diagnostic et résolution du problème INVALID_KMS_KEY

Pour diagnostiquer et traiter le code requis pour l'action INVALID_KMS_KEY, vous devez utiliser l'interface de AWS ligne de commande (CLI) et la console. AWS Key Management Service

Pour réactiver votre clé KMS

1. Appelez la méthode `DescribeBroker` pour récupérer l'identifiant `kmsKeyId` pour votre agent CMK.
2. Connectez-vous à la AWS Key Management Service console.
3. Sur la page Clés gérées par le client, recherchez l'ID de clé KMS de l'agent problématique et vérifiez que le statut est Activé.
4. Si votre clé KMS a été désactivée, réactivez-la en choisissant Actions de clé, puis choisissez Activer. Une fois votre clé réactivée, vous devez attendre que RabbitMQ retire l'agent de quarantaine.

Pour vérifier que les autorisations nécessaires sont toujours associées à la clé KMS du courtier, appelez la `ListGrant` ListGrant méthode pour vérifier cela `mq_rabbit_grant` et qu'`mq_grant`elles sont présentes. Si l'autorisation ou la clé KMS a été supprimée, vous devrez supprimer l'agent et en créer un nouveau avec toutes les autorisations nécessaires. Pour connaître la procédure de suppression d'un agent, consultez Suppression d'un agent.

Pour empêcher le code requis d'action critique INVALID_KMS_KEY, ne supprimez pas et ne désactivez pas manuellement une clé KMS ou une autorisation de clé CMK. Si vous souhaitez supprimer la clé, supprimez d'abord l'agent.

RabbitMQ sur Amazon MQ : alarme de limite de disque

L'alarme de limite de disque indique que le volume de disque utilisé par un nœud RabbitMQ a diminué en raison d'un nombre élevé de messages non consommés lors de l'ajout de nouveaux messages. RabbitMQ déclenche une alarme de limite de disque lorsque l'espace disque libre du courtier, identifié par la CloudWatch métrique Amazon`RabbitMQDiskFree`, atteint la limite de disque, identifiée par. `RabbitMQDiskFreeLimit` `RabbitMQDiskFreeLimit`est défini par Amazon MQ et a été défini en tenant compte de l'espace disque disponible pour chaque type d'instance de courtier.

Un courtier RabbitMQ sur Amazon MQ ayant déclenché une alarme de limite de disque ne sera plus disponible pour les nouveaux messages publiés. Si un éditeur et un consommateur sont connectés à la même connexion, le consommateur ne pourra pas non plus recevoir de messages. Lorsque RabbitMQ est exécuté dans un cluster, l'alarme de disque s'applique à l'ensemble du cluster. Si un nœud passe en dessous de cette limite, tous les autres nœuds bloqueront les messages entrants. En raison du manque d'espace disque, votre agent peut également rencontrer d'autres problèmes qui compliquent le diagnostic et la résolution de l'alarme.

Amazon MQ ne redémarre pas un agent confronté à une alarme de disque et retournera une exception pour les opérations API `RebootBroker` tant que l'agent continue de déclencher l'alarme.

> **Note**
>
> Vous ne pouvez pas rétrograder un agent d'un type d'instance `mq.m5` à un type d'instance `mq.t3.micro`. Si vous souhaitez rétrograder, vous devrez supprimer votre agent et en créer un nouveau.

Diagnostic et résolution des alarmes de limite de disque

Amazon MQ active les mesures pour votre agent par défaut. Vous pouvez consulter les statistiques de votre courtier en accédant à la CloudWatch console Amazon ou en utilisant l' CloudWatch API. `MessageCount`est une métrique utile pour diagnostiquer l'alarme de limite de disque RabbitMQ. Les messages sont stockés en mémoire jusqu'à ce qu'ils soient consommés ou rejetés. Un nombre élevé de messages indique une surutilisation de l'espace de stockage du disque et peut entraîner une alarme de disque.

Pour diagnostiquer l'alarme de limite de disque, utilisez la console de gestion Amazon MQ pour :

- Créez une nouvelle connexion pour utiliser les messages publiés dans les files d'attente.
- Purgez les messages des files d'attente.
- Supprimez les files d’attente de votre agent.

> **Note**
>
> Plusieurs heures peuvent être nécessaires pour que le statut de RABBITMQ_DISK_ALARM s'efface une fois que vous avez effectué les actions requises.

Pour éviter que l'alarme de limite de disque ne se reproduise, vous pouvez mettre à niveau votre type d'instance hôte vers une instance dotée de ressources supplémentaires. Pour plus d’informations sur la procédure de mise à jour du type d’instance de votre agent, consultez `UpdateBrokerInput` dans la Référence de l’API REST Amazon MQ. Nous vous recommandons également de maintenir vos éditeurs et vos consommateurs sur des connexions différentes.

Amazon MQ pour RabbitMQ : alarme de changement de type d'instance

`RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE`indique qu'un changement de type d'instance de courtier demandé ne peut pas être effectué en raison d'une utilisation élevée du disque sur le nœud RabbitMQ actuel. Amazon MQ pour RabbitMQ déclenche cette alarme lorsque l'utilisation actuelle du disque dépasse ce qui serait disponible sur le type d'instance demandé, tel qu'identifié par la métrique. CloudWatch `RabbitMQDiskFree`

Les courtiers RabbitMQ qui entrent dans `RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE` l'état continueront d'être disponibles pour vos applications, mais le changement de type d'instance demandé ne sera pas effectué. Amazon MQ autorise les redémarrages du broker dans cet état, mais vous ne pouvez pas modifier le type d'instance tant que l'utilisation du disque reste supérieure au seuil pour le type d'instance demandé. Le broker renverra une exception pour les opérations d'`ModifyBroker`API qui tentent de modifier le type d'instance dans cet état.

Diagnostic et adressage de l'alarme de changement de type d'instance

Amazon MQ active les mesures pour votre agent par défaut. Vous pouvez consulter les statistiques de votre courtier en accédant à la CloudWatch console ou en utilisant l' CloudWatch

API. `MessageCount`et `RabbitMQDiskFree` les métriques peuvent être utilisées pour diagnostiquer`RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE`.

Pour résoudre l'état de quarantaine et autoriser le changement de type d'instance, utilisez la console de gestion Amazon MQ pour :

- Créez une nouvelle connexion pour utiliser les messages publiés dans les files d'attente.
- Purgez les messages des files d'attente.
- Supprimez les files d’attente de votre agent.

> **Note**
>
> Plusieurs heures peuvent être nécessaires pour que le `RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE` statut soit effacé une fois que vous avez pris les mesures requises.

RabbitMQ sur Amazon MQ : IAM assume un rôle non valide

RabbitMQ sur Amazon MQ génère un code d'action critique INVALID_ASSUMEROLE requis lorsque l'ARN du rôle IAM spécifié dans n'est pas valide ou ne peut pas être assumé par Amazon MQ. `aws.arns.assume_role_arn` Cela peut se produire lorsque le rôle n'existe pas, se trouve sur un AWS compte différent de celui du courtier ou n'a pas la relation de confiance nécessaire avec mq.amazonaws.com.

Un courtier placé en quarantaine RABBITMQ_INVALID_ASSUMEROLE ne peut pas récupérer les informations d'identification ou les certificats requis pour l'authentification LDAP, ce qui rend l'authentification LDAP indisponible. Si LDAP est la seule méthode d'authentification configurée, les utilisateurs ne pourront pas se connecter au broker. Le rôle IAM est requis par Amazon MQ pour AWS accéder aux ressources référencées ARNs dans la configuration du broker, AWS Secrets Manager telles que les secrets ou les objets Amazon S3 utilisés pour l'authentification LDAP.

Diagnostic et adressage RABBITMQ_INVALID_ASSUMEROLE

Pour diagnostiquer et traiter le code requis pour l'action RABBITMQ_INVALID_ASSUMEROLE, vous devez utiliser Amazon Logs et la console. CloudWatch Gestion des identités et des accès AWS

Pour résoudre le problème d'attribution de rôle non valide

1. Accédez à Amazon CloudWatch Logs Insights et exécutez la requête suivante sur le groupe de journaux de votre courtier `/aws/amazonmq/broker/<broker-id>/general` :

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Recherchez les messages d'erreur similaires aux suivants :

```
[error] <0.254.0> aws_arn_config: {handle_assume_role,{error,
{assume_role_failed,"AWS service is unavailable"}}}
```

3. Vérifiez la configuration du rôle IAM et corrigez les problèmes tels que :
   - Assurez-vous que le rôle existe sur le même AWS compte que le courtier
   - Vérifiez que la politique de confiance autorise mq.amazonaws.com à assumer le rôle
   - Vérifiez que le rôle dispose des autorisations appropriées pour accéder aux AWS ressources requises
4. Validez le correctif à l'aide du point de terminaison de l'API de validation d'accès ARN avant de mettre à jour la configuration du broker.
5. Mettez à jour la configuration du courtier et redémarrez-le.

RabbitMQ sur Amazon MQ : ARN LDAP non valide

RabbitMQ sur Amazon MQ génère un code d'action critique INVALID_ARN_LDAP requis lorsque l'ARN configuré pour le mot de passe du compte de service LDAP n'est pas valide ou est inaccessible. Cela s'applique à ceux ARNs spécifiés dans `aws.arns.auth_ldap.dn_lookup_bind.password` ou`aws.arns.auth_ldap.other_bind.password`, qui doivent faire référence à des AWS Secrets Manager secrets contenant des mots de passe en texte clair.

Un courtier placé en quarantaine RABBITMQ_INVALID_ARN_LDAP ne peut pas s'authentifier auprès du compte de service LDAP, ce qui rend l'authentification LDAP indisponible. Si LDAP est la seule méthode d'authentification configurée, les utilisateurs ne pourront pas se connecter au broker. La non-validité ARNs peut être due à une syntaxe ARN mal formée, à des références à des secrets inexistants, à des secrets situés dans une AWS région différente de celle du broker ou à des GetSecretValue autorisations secretsmanager : insuffisantes dans le rôle IAM.

Diagnostic et adressage RABBITMQ_INVALID_ARN_LDAP

Pour diagnostiquer et traiter le code requis pour l'action RABBITMQ_INVALID_ARN_LDAP, vous devez utiliser Amazon Logs et la console. CloudWatch

Pour résoudre le problème d'ARN LDAP non valide

1. Accédez à Amazon CloudWatch Logs Insights et exécutez la requête suivante sur le groupe de journaux de votre courtier `/aws/amazonmq/broker/<broker-id>/general` :

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Recherchez les messages d'erreur similaires aux suivants :

```
[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. Vérifiez le secret du Gestionnaire de Secrets Manager et corrigez les problèmes tels que :

   - Vérifiez que le secret existe dans la même AWS région que le courtier
   - Vérifiez que la syntaxe de l'ARN est correcte
   - Assurez-vous que le rôle IAM possède les autorisations secretsmanager : GetSecretValue

4. Validez le correctif à l'aide du point de terminaison de l'API de validation d'accès ARN avant de mettre à jour la configuration du broker.

5. Mettez à jour la configuration du courtier et redémarrez-le.

RabbitMQ sur Amazon MQ : ARN HTTP non valide

RabbitMQ sur Amazon MQ génère un code d'action critique INVALID_ARN_HTTP requis lorsqu'un ou plusieurs certificats SSL ou fichier clé pour HTTP auth_backend ne sont ARNs pas valides ou inaccessibles. Cela s'applique aux ARNS spécifiés dans `aws.arns.auth_http.ssl_options.certfile` ou `aws.arns.auth_http.ssl_options.cacertfileaws.arns.auth_http.ssl_options.keyfile`, qui doivent faire référence à des objets et à des AWS Secrets Manager secrets Amazon S3 contenant des certificats et des clés privées.

Un courtier placé en quarantaine RABBITMQ_INVALID_ARN_HTTP ne peut pas s'authentifier via le serveur HTTP. Si le protocole HTTP est la seule méthode d'authentification configurée, les utilisateurs ne pourront pas se connecter au broker. La non-validité ARNs peut être due à une syntaxe ARN mal formée, à des références à des secrets inexistants, à des secrets situés dans une AWS région différente de celle du broker ou à des GetSecretValue autorisations s3 : GetObject / secretsmanager : insuffisantes dans le rôle IAM.

Diagnostic et adressage RABBITMQ_INVALID_ARN_HTTP

Pour diagnostiquer et traiter le code requis pour l'action RABBITMQ_INVALID_ARN_HTTP, vous devez utiliser Amazon Logs et la console. CloudWatch

Pour résoudre le problème d'ARN HTTP non valide

1. Accédez à Amazon CloudWatch Logs Insights et exécutez la requête suivante sur le groupe de journaux de votre courtier `/aws/amazonmq/broker/<broker-id>/general` :

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Recherchez les messages d'erreur similaires aux suivants :

```
[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. Vérifiez le secret de S3 Object/Secrets Manager et corrigez les problèmes tels que :
   - Vérifiez que la ressource existe dans la même AWS région que le courtier
   - Vérifiez que la syntaxe de l'ARN est correcte
   - Assurez-vous que le rôle IAM dispose des autorisations s3 : GetObject et secretsmanager : GetSecretValue
4. Validez le correctif à l'aide du point de terminaison de l'API de validation d'accès ARN avant de mettre à jour la configuration du broker.
5. Mettez à jour la configuration du courtier et redémarrez-le.

RabbitMQ sur Amazon MQ : ARN SSL non valide

RabbitMQ sur Amazon MQ génère un code d'action critique INVALID_ARN_SSL requis lorsqu'un ou plusieurs certificats CA Truststore pour EXTERNAL auth_mechanism ne sont pas valides ou ARNs inaccessibles. Cela s'applique à l'ARNS spécifié dans `aws.arns.ssl_options.cacertfile` ou`aws.arns.management.ssl.cacertfile`, qui doit faire référence à l'objet Amazon S3 ou ACM PCA contenant le certificat.

Un courtier placé en quarantaine RABBITMQ_INVALID_ARN_SSL ne peut pas authentifier les certificats clients lors de connexions TLS mutuelles car aucun truststore valide n'est configuré. Si le mécanisme d'authentification EXTERNE est la seule méthode d'authentification configurée, les utilisateurs ne pourront pas se connecter au courtier. La non-validité ARNs peut être due à une syntaxe ARN mal formée, à des références à des objets S3 inexistants, à des objets S3 situés dans une AWS région différente de celle du broker ou à des autorisations s3 : GetObject /acm-pca : GetCertificateAuthorityCertificate insuffisantes dans le rôle IAM.

Diagnostic et adressage RABBITMQ_INVALID_ARN_SSL

Pour diagnostiquer et traiter le code requis pour l'action RABBITMQ_INVALID_ARN_SSL, vous devez utiliser Amazon Logs et la console. CloudWatch

Pour résoudre le problème d'ARN SSL non valide

1. Accédez à Amazon CloudWatch Logs Insights et exécutez la requête suivante sur le groupe de journaux de votre courtier `/aws/amazonmq/broker/<broker-id>/general` :

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Recherchez les messages d'erreur similaires aux suivants :

```
[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. Vérifiez l'objet S3/ACM-PCA et corrigez les problèmes tels que :

   - Vérifiez que le secret existe dans la même AWS région que le courtier
   - Vérifiez que la syntaxe de l'ARN est correcte
   - Assurez-vous que le rôle IAM dispose des autorisations s3 : GetObject /acm-pca : GetCertificateAuthorityCertificate

4. Validez le correctif à l'aide du point de terminaison de l'API de validation d'accès ARN avant de mettre à jour la configuration du broker.
5. Mettez à jour la configuration du courtier et redémarrez-le.

RabbitMQ sur Amazon MQ : ARN non valide

RabbitMQ sur Amazon MQ génère un code d'action critique INVALID_ARN requis lorsqu'une ou plusieurs configurations ARNs configurées dans le broker ne sont pas valides ou inaccessibles. Cela s'applique aux certificats SSL, ARNs aux AWS Secrets Manager secrets, aux objets Amazon S3 ou à d'autres références de AWS ressources non couvertes par des codes de quarantaine plus spécifiques tels que RABBITMQ_INVALID_ARN_LDAP ou RABBITMQ_INVALID_ASSUMEROLE.

Un broker placé en quarantaine par RABBITMQ_INVALID_ARN peut voir ses fonctionnalités dégradées en fonction de celles qui ne sont pas valides. ARNs Les fonctionnalités qui dépendent de ressources inaccessibles ne seront pas disponibles et le courtier enregistrera les erreurs indiquant quel ARN n'a pas pu être résolu. L'impact sur la disponibilité des courtiers dépend de la nécessité ou non d'utiliser un ARN non valide pour les opérations critiques des courtiers.

Diagnostic et adressage RABBITMQ_INVALID_ARN

Pour diagnostiquer et traiter le code requis pour l'action RABBITMQ_INVALID_ARN, vous devez utiliser Amazon CloudWatch Logs et la console de service appropriée AWS pour la ressource affectée.

Pour résoudre le problème d'ARN non valide

1. Accédez à Amazon CloudWatch Logs Insights et exécutez la requête suivante sur le groupe de journaux de votre courtier `/aws/amazonmq/broker/<broker-id>/general` :

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Recherchez les messages d'erreur similaires aux suivants :

```
[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. Consultez la AWS ressource et corrigez les problèmes tels que :

   - Vérifiez que la ressource existe dans la même AWS région que le courtier
   - Vérifiez que la syntaxe de l'ARN est correcte
   - Assurez-vous que le rôle IAM dispose des autorisations appropriées pour accéder à la ressource

4. Validez le correctif à l'aide du point de terminaison de l'API de validation d'accès ARN avant de mettre à jour la configuration du broker.
5. Mettez à jour la configuration du courtier et redémarrez-le.

Ressources connexes

Ressources Amazon MQ

Le tableau suivant répertorie les ressources utiles pour travailler avec Amazon MQ.

Ressource	Description
Référence de l'API REST Amazon MQ	Descriptions des ressources REST, exemples de demande, méthodes HTTP, schémas, paramètres et erreurs renvoyées par le service.
Amazon MQ dans le manuel de référence des commandes AWS CLI	Descriptions des AWS CLI commandes que vous pouvez utiliser pour travailler avec les courtiers de messages.
Amazon MQ dans le guide de l'utilisateur AWS CloudFormation	La ressource `AWS::AmazonMQ::Broker` vous permet de créer des agents Amazon MQ, d'ajouter des modifications de configuration ou de modifier des utilisateurs pour l'agent spécifié, de renvoyer des informations sur l'agent spécifié et de supprimer l'agent spécifié. La ressource `AWS::AmazonMQ::Configuration` vous permet de créer des configurations Amazon MQ, d'ajouter des modifications de configuration ou de modifier des utilisateurs et de renvoyer des informations sur la configuration spécifiée.
Régions et points de terminaison	Informations sur les régions et les points de terminaison Amazon MQ
Page produit	Page web principale pour de plus amples informations sur Amazon MQ.

Ressource	Description
Forum de discussion	Un forum communautaire pour les développeurs où ils peuvent discuter des questions techniques liées à Amazon MQ.
AWS Informations sur le Support Premium	La page Web principale contenant des informations sur le support AWS Premium one-on-one, un canal d'assistance rapide destiné à vous aider à créer et à exécuter des applications sur des services AWS d'infrastructure

Ressources Amazon MQ for ActiveMQ

Le tableau suivant répertorie les ressources utiles pour travailler avec Apache ActiveMQ.

Ressource	Description
Apache ActiveMQ Getting Started Guide	Documentation officielle d'Apache ActiveMQ.
ActiveMQ in Action	Guide d'Apache ActiveMQ qui couvre l'anatomie des messages JMS, les connecteurs, la persistance des messages, l'authentification et les autorisations.
Clients inter-langages	Liste de langages de programmation et des bibliothèques Apache ActiveMQ correspondantes. Voir également Client ActiveMQ et Client QpidJMS.

Ressources Amazon MQ for RabbitMQ

Le tableau suivant répertorie les ressources utiles pour travailler avec RabbitMQ.

Ressource	Description
Le guide de démarrage de RabbitMQ	Documentation officielle de RabbitMQ.
Bibliothèques clientes et outils de développement RabbitMQ	Guide sur les bibliothèques client officiellement prises en charge et les outils de développeur pour utiliser RabbitMQ à l'aide d'une grande variété de langages de programmation et de plateformes.
Meilleures pratiques de RabbitMQ	Guide de CloudAMQP sur les bonnes pratiques et recommandations pour utiliser RabbitMQ.

Notes de mise à jour Amazon MQ

Le tableau suivant répertorie les lancements de fonctions et les améliorations apportées à Amazon MQ.

Date	Mise à jour de la documentation
19 février 2026	Amazon MQ Amazon MQ prend désormais en charge ActiveMQ 5.19, une nouvelle version mineure du moteur. Pour plus d'informations, veuillez consulter la rubrique • Page de mise à jour d'ActiveMQ 5.19 • Gestion des versions du moteur Amazon MQ for ActiveMQ • Mise à niveau d'une version du moteur d'agent Amazon MQ • Utilisation des fichiers de configuration XML Spring
22 janvier 2026	Amazon MQ prend désormais en charge le plug-in d'échange de sujets JMS pour les courtiers utilisant RabbitMQ 4.2 et versions ultérieures. Vous pouvez utiliser le client JMS RabbitMQ officiel pour exécuter des charges de travail JMS sur Amazon MQ pour le courtier RabbitMQ. Il supporte JMS 1.1, 2.0 et 3.1. Pour plus d'informations, veuillez consulter la rubrique • Spécification officielle de JMS 2.0 (rétrocompatible et étendue avec JMS 1.1) • Spécification officielle du JMS 3.1 • Limitation du client JMS RabbitMQ • Connexion de votre application JMS au courtier Amazon MQ pour RabbitMQ
8 janvier 2026	Amazon MQ prend désormais en charge l'authentification par certificat SSL pour les courtiers sur RabbitMQ 4.2 et versions ultérieures à l'aide de certificats clients X.509 et d'une configuration TLS mutuelle (MTLS). Vous pouvez configurer l'authentification par certificat SSL et le protocole MTL via AWS

Date	Mise à jour de la documentation
	Management Console AWS CloudFormation AWS CLI,, ou AWS CDK partout Régions AWS où Amazon MQ est disponible. Pour plus d'informations, consultez Authentification par certificat SSL et Configuration des mTLS.
6 janvier 2026	Amazon MQ prend désormais en charge l'authentification et l'autorisation HTTP pour les courtiers utilisant RabbitMQ 4.2 et versions ultérieures avec des serveurs HTTP externes. Vous pouvez configurer l'authentification HTTP via AWS Management Console AWS CloudFormation, AWS CLI, ou AWS CDK partout Régions AWS où Amazon MQ est disponible. Pour plus d'informations, consultez Authentification et autorisation HTTP.
20 novembre 2025	Amazon MQ prend désormais en charge RabbitMQ 4.2, une nouvelle version majeure qui introduit la prise en charge native du protocole AMQP 1.0, un nouveau magasin de métadonnées Khepri basé sur Raft, des pelles locales et des priorités de messages pour les files d'attente du quorum. RabbitMQ 4.2 inclut également diverses corrections de bogues et améliorations des performances pour la gestion du débit et de la mémoire. Bien que cette version introduise de nouvelles fonctionnalités, il y a quelques changements importants. Pour plus d’informations, veuillez consulter la rubrique • Lapin MQ 4 • Notes de mise à jour de RabbitMQ open source • Configuration des limites de ressources • Protocoles pris en charge • Mises à niveau de la version Amazon MQ

Date	Mise à jour de la documentation
18 novembre 2024	Amazon MQ prend désormais en charge les instances m7g alimentées par Graviton3 pour RabbitMQ dans une gamme de tailles allant de taille moyenne à 16 fois grande en Afrique (Le Cap). Pour de plus amples informations, veuillez consulter Types d'instances de courtier Amazon MQ pour RabbitMQ.
17 novembre 2025	Amazon MQ prend désormais en charge l'authentification et l'autorisation LDAP pour les courtiers RabbitMQ avec des services d'annuaire LDAP externes. Vous pouvez configurer LDAP via AWS Management Console AWS CloudFormation, AWS CLI, ou partout AWS CDK Régions AWS où Amazon MQ est disponible. Pour de plus amples informations, veuillez consulter Authentification et autorisation LDAP pour Amazon MQ pour RabbitMQ.
22 octobre 2025	Amazon MQ est désormais disponible dans la région Asie-Pacifique (Nouvelle Zélande). Pour des informations sur les Régions, consultez Régions AWS et points de terminaison dans le Guide de référence général d'AWS .
3 septembre 2025	Amazon MQ prend désormais en charge l'authentification et l'autorisation OAuth 2.0 pour les courtiers RabbitMQ auprès de fournisseurs d'identit é publics (). IdPs Vous pouvez configurer la OAuth version 2.0 via AWS Management Console AWS CloudFormation AWS CLI,, ou AWS CDK partout Régions AWS où Amazon MQ est disponible. Pour de plus amples informations, veuillez consulter OAuth Authentification et autorisation 2.0 pour Amazon MQ pour RabbitMQ.

Date	Mise à jour de la documentation
22 juillet 2025	Amazon MQ prend désormais en charge les `m7g` instances basées sur Graviton3 pour RabbitMQ dans une gamme de tailles allant de taille moyenne à 16 fois grande. Les clusters RabbitMQ exécutés sur des `m7g` instances offrent une capacité de charge de travail jusqu'à 50 % supérieure et jusqu'à 85 % d'amélioration du débit par rapport à Amazon MQ pour les clusters RabbitMQ comparables exécutés sur des instances. `m5` `M7g`les instances ont également des tailles de volume de disque optimisées qui varient en fonction de la taille de l'instance. Pour de plus amples informati ons, veuillez consulter Broker instance types. `M7g`les instances sur Amazon MQ sont disponibles aujourd'hui dans toutes les régions généralement disponibles, à l'exception de l'Afrique (Le Cap), du Canada-Ouest (Calgary) et de l'Europe (Milan).
8 juillet 2025	Amazon MQ est désormais disponible dans la région Asie-Pacifique (Taipei). Pour des informations sur les Régions, consultez Régions AWS et points de terminaison dans le Guide de référence général d'AWS .
22 avril 2025	Vous pouvez désormais supprimer les configurations de broker Amazon MQ à l'aide de l'`DeleteConfiguration` API. Pour plus d'informations, consultez Configurations dans le manuel Amazon MQ API Reference.
16 avril 2025	Amazon MQ pour RabbitMQ prend désormais en charge l'utilisation de points de terminaison à double pile (IPv4 et IPv6) pour se connecter à des courtiers publics et privés. Pour plus d'informations, consultez Connecting to Amazon MQ et Configuring a private Amazon MQ broker.
7 avril 2025	Amazon MQ est désormais disponible dans les régions Asie-Pacifique (Thaïlande) et Mexique (centre). Pour des informations sur les Régions, consultez Régions AWS et points de terminaison dans le Guide de référence général d'AWS .

Date	Mise à jour de la documentation
13 février 2025	Les points de terminaison FIPS de l'API Amazon MQ sont désormais disponibles dans les régions du Canada (Centre) et du Canada Ouest (Calgary). Pour plus d'informations sur l'utilisation des points de terminaison FIPS avec l'API Amazon MQ, consultez. Connecting to Amazon MQ Pour des informations sur les Régions, consultez Régions AWS et points de terminaison dans le Guide de référence général d'AWS .
12 février 2025	Amazon MQ annonce les dates de fin de support du type d'instance suivantes : Broker instance types • `mq.t2.micro` ActiveMQ : 12 mai 2025 • `mq.m4.large` ActiveMQ : 12 mai 2025 Vous ne pouvez pas créer de `mq.t2.micro` courtiers à `mq.m4.large` compter du 17 mars 2025.
10 décembre 2024	Amazon MQ prend désormais en charge l'utilisation AWS PrivateLink pour vous connecter entre vos clouds privés virtuels (VPCs) et l'API Amazon MQ sans exposer votre trafic à l'Internet public. Pour de plus amples informations, veuillez consulter the section called “Connectez-vous à Amazon MQ à l'aide de AWS PrivateLink”.
18 novembre 2024	Amazon MQ est désormais disponible dans la région Asie-Pacifique (Malaisie). Pour des informations sur les Régions, consultez Régions AWS et points de terminaison dans le Guide de référence général d'AWS .

Date	Mise à jour de la documentation
14 novembre 2024	Amazon MQ annonce les dates de fin de support des versions du moteur suivantes : Gestion des versions du moteur Amazon MQ for ActiveMQ • ActiveMQ 5.17 : 16 juin 2025 Gestion des versions du moteur Amazon MQ for RabbitMQ • RabbitMQ 3.11 : 17 février 2025 • RabbitMQ 3.12 : 17 mars 2025 Pour plus d'informations sur la mise à niveau vers la dernière version, voir Mise à niveau d'une version du moteur d'agent Amazon MQ
13 novembre 2024	Amazon MQ prend désormais en charge les points de terminaison de service à double pile auxquels vous pouvez vous connecter à l'aide de l'un ou de l'autre. IPv4 IPv6 Les points de terminaison de service régionaux à double pile Amazon MQ peuvent être résolus à la fois A avec des enregistrements DNS. AAAA Pour de plus amples informations, veuillez consulter ???.
25 juillet 2024	Amazon MQ prend désormais en charge ActiveMQ 5.18, une nouvelle version mineure du moteur. Pour plus d'informations, consultez les ressources suivantes : • Page de mise à jour d'ActiveMQ 5.18 • Gestion des versions du moteur Amazon MQ for ActiveMQ • Mise à niveau d'une version du moteur d'agent Amazon MQ • Utilisation des fichiers de configuration XML Spring

Date	Mise à jour de la documentation
22 juillet 2024	Amazon MQ prend désormais en charge les files d'attente de quorum uniquement pour les courtiers utilisant la version 3.13 ou supérieure. Les files d'attente de quorum sont un type de file FIFO répliqué qui utilise l'algorithme de consensus Raft pour maintenir la cohérence des données. Les files d'attente du quorum permettent de gérer les messages toxiques, ce qui peut vous aider à gérer les messages non traités. Pour commencer à utiliser les files d'attente de quorum, voirFiles d'attente de quorum pour RabbitMQ sur Amazon MQ.
2 juillet 2024	Amazon MQ pour RabbitMQ prend désormais en charge RabbitMQ 3.13, une version mineure. Pour tous les courtiers utilisant la version 3.13 du moteur ou une version ultérieure, Amazon MQ gère les mises à niveau vers la dernière version de correctif prise en charge pendant la période de maintenance. Pour de plus amples informations, veuillez consulter Mise à niveau d'une version du moteur d'agent Amazon MQ. Directives de dimensionnement d'Amazon MQ pour RabbitMQont été mis à jour pour inclure de nouvelles limites pour les files d'attente, les consommateurs par canal et les pelles pour les courtiers utilisant la version 3.13 du moteur. Pour plus d'informations sur les correctifs et les fonctionnalités de cette version, consultez les notes de version de RabbitMQ 3.13 sur le référentiel du serveur RabbitMQ. GitHub Pour plus d’informations sur les versions et les mises à niveau d’agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.
10 juin 2024	Amazon MQ est désormais disponible dans la région Ouest du Canada (Calgary). Pour des informations sur les Régions, consultez Régions AWS et points de terminaison dans le Guide de référence général d'AWS .

Date	Mise à jour de la documentation
10 mai 2024	Le calendrier de support de la version Amazon MQ indique le moment où le support d'une version du moteur de courtage atteint la fin du support. Lorsque le support d'une version du moteur atteint la fin du support, Amazon MQ informe automatiquement tous les courtiers de la version vers la prochaine version mineure prise en charge. Amazon MQ fournit un préavis d'au moins 90 jours avant la fin du support d'une version du moteur. Pour consulter le calendrier de support des versions et la fin du support, consultez ce qui suit : • Gestion des versions du moteur Amazon MQ for ActiveMQ • Gestion des versions du moteur Amazon MQ for RabbitMQ Vous pouvez également activer les mises à niveau automatiques des versions mineures pour que votre courtier passe à la version de correctif suivante pendant une période de maintenance. Pour de plus amples informations, consultez Mise à niveau d'une version du moteur d'agent Amazon MQ.

Date	Mise à jour de la documentation
9 mai 2024	Amazon MQ pour RabbitMQ prend désormais en charge RabbitMQ 3.12, une version mineure. Tous les courtiers du 3.12.13 et des versions ultérieur es utilisent Classic Queues version 2 (CQv2), et toutes les files d'attente du 3.12.13 et des versions ultérieures se comportent comme des files d'attente paresseuses. Nous recommandons aux courtiers utilisant les versions antérieures à la version 3.12.13 d'activer CQv2 et de suspendre les files d'attente, ou de passer à la dernière version d'Amazon MQ pour RabbitMQ. Pour plus d’informations sur les correctifs et les fonctionnalités de cette version, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.12 sur le référentiel du serveur RabbitMQ. GitHub Pour plus d’informations sur les versions et les mises à niveau d’agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.
4 mars 2024	Amazon MQ pour RabbitMQ prend désormais en charge RabbitMQ 3.11.28. Pour plus d’informations sur les correctifs et les fonctionnalités de cette version, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.11.28 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ Pour plus d’informations sur les versions et les mises à niveau d’agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.

Date	Mise à jour de la documentation
19 janvier 2024	Amazon MQ pour RabbitMQ ne prend pas en charge le nom d'utilisateur « invité » et supprimera le compte invité par défaut lorsque vous créerez un nouveau courtier. Amazon MQ supprimera également régulièrement tout compte créé par un client appelé « invité ».
15 décembre 2023	Amazon MQ est désormais disponible dans la région Israël (Tel Aviv). Pour des informations sur les Régions, consultez Régions AWS et points de terminaison dans le Guide de référence général d'AWS .
11 décembre 2023	Amazon MQ for RabbitMQ prend désormais en charge RabbitMQ 3.10.25. Pour plus d'informations sur les correctifs et les fonctionnalités de cette version, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.10.25 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ Pour plus d'informations sur les versions et les mises à niveau d'agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.
26 octobre 2023	Amazon MQ a publié les dernières versions mineures d'ActiveMQ 5.15.16, 5.16.7, 5.17.6 avec une mise à jour critique. Nous avons rendu obsolètes les anciennes versions mineures d'ActiveMQ et mettrons à jour tous les agents sur toutes les versions de 5.15 à 5.15.16, de 5.16 à 5.16.7 et de 5.17 à 5.17.6. Pour en savoir plus sur la mise à jour de votre agent ActiveMQ, consultez Gestion des versions du moteur Amazon MQ for ActiveMQ.

Date	Mise à jour de la documentation
27 septembre 2023	Amazon MQ for RabbitMQ prend désormais en charge RabbitMQ 3.11.20. Pour plus d'informations sur les correctifs et les fonctionnalités de cette version, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.11.20 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ Pour plus d'informations sur les versions et les mises à niveau d'agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.
17 juillet 2023	Amazon MQ for RabbitMQ prend désormais en charge RabbitMQ 3.11.16. Pour plus d'informations sur les correctifs et les fonctionnalités de cette version, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.11.16 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ Pour plus d'informations sur les versions et les mises à niveau d'agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.
17 juillet 2023	Amazon MQ for RabbitMQ prend désormais en charge la création de configurations et leur application à votre agent RabbitMQ. Pour en savoir plus sur l'ajout de configurations à votre agent, consultez RabbitMQ Broker Configurations. Pour en savoir plus sur cette fonctionnalité, consultez : • Politiques d'opérateur • Changements concernant les politiques d'opérateur

Date	Mise à jour de la documentation
23 juin 2023	Amazon MQ prend désormais en charge ActiveMQ 5.17.3, une nouvelle version mineure du moteur. Cette version prend en charge la nouvelle fonctionnalité de réplication de données entre régions (CRDR) d'Amazon MQ. Pour plus d'informations, consultez les ressources suivantes : • Pour bien démarrer avec la réplication CRDR, consultez Réplication de données entre régions pour Amazon MQ pour ActiveMQ dans le Guide du développeur. • Notes de mise à jour d'ActiveMQ 5.17.3 • Gestion des versions du moteur Amazon MQ for ActiveMQ • Mise à niveau d'une version du moteur d'agent Amazon MQ • Utilisation des fichiers de configuration XML Spring
21 juin 2023	Amazon MQ pour ActiveMQ propose désormais une fonctionnalité de réplication de données entre régions (CRDR) qui permet la réplication asynchrone des messages depuis le courtier principal d'une région principale vers le courtier répliqué d'une région de réplication. AWS Si l'agent principal situé dans la région principale échoue, vous pouvez promouvoir l'agent de répliques situé dans la région secondaire au rang d'agent principal en lançant une commutation ou un basculement. Pour bien démarrer avec la réplication CRDR, consultez Réplication de données entre régions pour Amazon MQ pour ActiveMQ dans le Guide du développeur.

Date	Mise à jour de la documentation
18 mai 2023	Amazon MQ est désormais disponible dans les régions suivantes : • Asie-Pacifique (Melbourne) • Asie-Pacifique (Hyderabad) • Europe (Espagne) • Europe (Zurich) Pour des informations sur les Régions, consultez Régions AWS et points de terminaison dans le Guide de référence général d'AWS .
14 avril 2023	Amazon MQ for RabbitMQ prend désormais en charge la version RabbitMQ 3.9.27. Pour plus d'informations sur les correctifs et les fonctionnalités de cette version, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.9.27 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ Pour plus d'informations sur les versions et les mises à niveau d'agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.

Date	Mise à jour de la documentation
14 avril 2023	Amazon MQ pour RabbitMQ prend désormais en charge RabbitMQ version 3.10.20. Pour plus d'informations sur les correctifs et les fonctionnalités de cette version, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.10.20 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ Pour plus d'informations sur les versions et les mises à niveau d'agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.
31 mars 2023	Amazon MQ pour RabbitMQ a désactivé le moteur RabbitMQ version 3.10.17 L'équipe Amazon MQ pour RabbitMQ et les responsables open source de RabbitMQ ont identifié un problème lié à la console de gestion RabbitMQ dans la version 3.10.17. Amazon MQ a retiré cette version. Pour atténuer l'impact de ce problème, créez de nouveaux agents avec la version 3.10.10 pendant que nous travaillons à une nouvelle version corrective de RabbitMQ. Nous vous recommandons d'activer l'option de mise à niveau de version pour obtenir automatiquement les dernières corrections de bogues, mises à jour de sécurité et améliorations de performances. Pour plus d'informations sur les versions disponibles d'Amazon MQ pour RabbitMQ, consultez Versions du moteur Amazon MQ pour RabbitMQ.

Date	Mise à jour de la documentation
1er mars 2023	Amazon MQ pour RabbitMQ prend désormais en charge RabbitMQ version 3.10.17. Pour plus d'informations sur les correctifs et les fonctionnalités de cette version, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.10.17 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ Pour plus d'informations sur les versions et les mises à niveau d'agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.
21 février 2023	Amazon MQ pour RabbitMQ s'intègre désormais à AWS Key Managemen t Service (KMS) pour proposer un chiffrement côté serveur. Vous pouvez désormais sélectionner votre propre clé CMK gérée par le client ou utiliser une clé KMS AWS gérée dans votre AWS KMS compte. Pour de plus amples informations, veuillez consulter Chiffrement au repos. Amazon MQ prend en charge l'utilisation AWS KMS des clés des manières suivantes. • Clé KMS détenue par Amazon MQ (valeur par défaut) – La clé est détenue par Amazon MQ et ne figure pas dans votre compte. • AWS clé KMS AWS gérée : la clé KMS gérée (`aws/mq`) est une clé KMS de votre compte créée, gérée et utilisée en votre nom par Amazon MQ. • Sélection d'une clé KMS existante gérée par le client – Vous créez et gérez les clés KMS gérées par le client dans AWS Key Management Service (KMS).

Date	Mise à jour de la documentation
13 janvier 2023	Amazon MQ for RabbitMQ prend désormais en charge RabbitMQ version 3.8.34. Pour plus d'informations sur les correctifs et les fonctionnalités de cette version, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.8.34 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ Pour plus d'informations sur les versions et les mises à niveau d'agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.
15 décembre 2022	Amazon MQ for RabbitMQ prend désormais en charge la version RabbitMQ 3.9.24. Pour plus d'informations sur les correctifs et les fonctionnalités de cette version, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.9.24 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ Pour plus d'informations sur les versions et les mises à niveau d'agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.
13 décembre 2022	Amazon MQ est désormais disponible dans la région du Moyen-Orient (UAE). Pour des informations sur les Régions, consultez Régions AWS et points de terminaison dans le Guide de référence général d'AWS .

Date	Mise à jour de la documentation
14 novembre 2022	Amazon MQ for RabbitMQ prend désormais en charge la version 3.10, une version majeure du moteur. Vous pouvez désormais activer la version 2 (CQv2) des files d'attente classiques sur vos files d'attente RabbitMQ. Les mises à jour directes des versions 3.8 à 3.10 ne sont pas prises en charge. Pour plus d'informations, consultez les ressources suivantes : • Notes de mise à jour de RabbitMQ 3.10.10 • Journal des modifications RabbitMQ Pour plus d'informations sur les versions et les mises à niveau d'agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.
9 novembre 2022	Amazon MQ prend désormais en charge ActiveMQ 5.17.2, une nouvelle version mineure du moteur. Pour plus d'informations, consultez les ressources suivantes : • Notes de mise à jour d'ActiveMQ 5.17.2 • Gestion des versions du moteur Amazon MQ for ActiveMQ • Mise à niveau d'une version du moteur d'agent Amazon MQ • Utilisation des fichiers de configuration XML Spring
17 août 2022	Amazon MQ prend désormais en charge ActiveMQ 5.17.1, une nouvelle version majeure du moteur. Pour plus d'informations, consultez les ressources suivantes : • Notes de mise à jour d'ActiveMQ 5.17.1 • Gestion des versions du moteur Amazon MQ for ActiveMQ • Mise à niveau d'une version du moteur d'agent Amazon MQ • Utilisation des fichiers de configuration XML Spring

Date	Mise à jour de la documentation
14 juillet 2022	Amazon MQ prend désormais en charge ActiveMQ 5.16.5, une version mineure du moteur. Pour plus d'informations, consultez les ressources suivantes : • Notes de mise à jour d'ActiveMQ 5.16.5 • Gestion des versions du moteur Amazon MQ for ActiveMQ • Utilisation des fichiers de configuration XML Spring • Mise à niveau d'une version du moteur d'agent Amazon MQ
4 mai 2022	Amazon MQ ajoute un langage inclusif pour l'élément `networkConnector` dans la configuration de l'agent. • Création et configuration d'un réseau d'agents Amazon MQ
25 avril 2022	Amazon MQ Cette version ajoute l'état d'agent `CRITICAL_ACTION_REQUIRED` et la propriété d'API `ActionRequired` . `CRITICAL_ACTION_REQUIRED` vous informe lorsque votre courtier est dégradé. `ActionRequired` vous fournit un code que vous pouvez utiliser pour trouver des instructions dans le Guide du développeur sur la façon de résoudre le problème. • Résolution des problèmes • Documentation `ActionRequired` dans la Référence de l'API Amazon MQ.
20 avril 2022	Amazon MQ prend désormais en charge ActiveMQ 5.16.4, une version mineure du moteur. Pour plus d'informations, consultez les ressources suivantes : • Notes de mise à jour d'ActiveMQ 5.16.4 • Gestion des versions du moteur Amazon MQ for ActiveMQ • Utilisation des fichiers de configuration XML Spring • Mise à niveau d'une version du moteur d'agent Amazon MQ

Date	Mise à jour de la documentation
1er mars 2022	Amazon MQ est désormais disponible dans la région Asie-Pacifique (Jakarta) . Pour des informations sur les Régions, consultez Régions AWS et points de terminaison dans le Guide de référence général d'AWS .
25 février 2022	Amazon MQ for RabbitMQ prend désormais en charge RabbitMQ version 3.8.27. Pour plus d'informations sur les correctifs et les fonctionnalités de cette version, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.8.27 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ Pour plus d'informations sur les versions et les mises à niveau d'agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.
16 février 2022	Amazon MQ est désormais disponible dans la Région Afrique (Le Cap). Pour des informations sur les Régions, consultez Régions AWS et points de terminaison dans le Guide de référence général d'AWS .

Date	Mise à jour de la documentation
14 février 2022	Amazon MQ for RabbitMQ prend désormais en charge la version RabbitMQ 3.9.13. Mises à niveau automatiques des versions mineures ne peut pas être utilisé pour passer de Rabbit 3.8 à 3.9. Pour ce faire, mettez à niveau votre agent manuellement. Pour plus d'informations sur les nouvelles fonctionnalités introduites dans RabbitMQ 3.9, consultez la page des notes de publication de la version 3.9.0 sur le site Web. GitHub **Note** Actuellement, Amazon MQ ne prend pas en charge des flux, ni l'utilisation de la journalisation structurée dans JSON, introduite dans RabbitMQ 3.9. Pour plus d’informations sur les correctifs et les fonctionnalités de cette version, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.9.13 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ Pour plus d’informations sur les versions et les mises à niveau d’agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.
7 février 2022	Amazon MQ for RabbitMQ introduit de nouvelles mesures d'agent, ce qui vous permet de surveiller l'utilisation moyenne des ressources sur les trois nœuds d'un déploiement de cluster. Pour plus d’informations, consultez les ressources suivantes : • the section called “Métriques pour RabbitMQ”

Date	Mise à jour de la documentation
18 janvier 2022	Amazon MQ for RabbitMQ prend désormais en charge RabbitMQ version 3.8.26. Pour plus d'informations sur les correctifs et les fonctionnalités de cette version, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.8.26 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ Pour plus d'informations sur les versions et les mises à niveau d'agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.
13 janvier 2022	Amazon MQ introduit le code d'état `RABBITMQ_MEMORY_ALARM` pour vous informer lorsque votre agent a déclenché une alarme de mémoire élevée et qu'il se trouve dans un état malsain. Amazon MQ fournit des informations détaillées et des recommandations pour vous aider à diagnostiquer, résoudre et prévenir les alarmes de mémoire élevée. Pour plus d'informations, consultez les rubriques suivantes. • the section called " `RABBITMQ_MEMORY_ALARM` "
6 janvier 2022	Lorsque vous configurez CloudWatch Logs for Amazon MQ pour les courtiers ActiveMQ, Amazon MQ prend en charge l'utilisation des clés contextuelles `aws:SourceAccount` et des conditions globales dans les politiques basées sur `aws:SourceArn` les ressources IAM afin d'éviter le problème de confusion des adjoints. Pour plus d'informations, consultez les rubriques suivantes. • the section called "Prévention du cas de figure de l'adjoint désorienté entre services"

Date	Mise à jour de la documentation
20 décembre 2021	Amazon MQ pour ActiveMQ introduit un ensemble de nouvelles mesures, vous permettant de surveiller le nombre maximal de connexions que vous pouvez établir avec votre agent à l'aide de différents protocoles de transport pris en charge, ainsi qu'une nouvelle mesure supplémentaire qui vous permet de surveiller le nombre de nœuds connectés à votre agent dans unréseau d'agents. Pour plus d'informations, consultez les rubriques suivantes. • the section called "Métriques pour ActiveMQ"
16 novembre 2021	Amazon MQ for RabbitMQ prend désormais en charge RabbitMQ version 3.8.23. Pour plus d'informations sur les correctifs et les fonctionnalités de cette version, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.8.23 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ Pour plus d'informations sur les versions et les mises à niveau d'agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ.
12 octobre 2021	Amazon MQ prend désormais en charge ActiveMQ 5.16.3, une version mineure du moteur. Pour plus d'informations, consultez les ressources suivantes : • Notes de mise à jour d'ActiveMQ • Gestion des versions du moteur Amazon MQ for ActiveMQ • Mise à niveau d'une version du moteur d'agent Amazon MQ • Utilisation des fichiers de configuration XML Spring

Date	Mise à jour de la documentation
8 septembre 2021	Amazon MQ for RabbitMQ prend désormais en charge RabbitMQ version 3.8.22. Cette version inclut un correctif pour un problème avec les files d'attente utilisant TTL par message (time to live), identifié dans la version précédemm ent prise en charge, RabbitMQ 3.8.17. Nous vous recommandons de mettre à niveau vos agents existants vers la version 3.8.22. Pour plus d'informations sur les correctifs et les fonctionnalités de cette version, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.8.22 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ Pour plus d'informations sur les versions et les mises à niveau d'agent Amazon MQ for RabbitMQ prises en charge, consultez Gestion des versions du moteur Amazon MQ for RabbitMQ
25 août 2021	Amazon MQ pour RabbitMQ a temporairement désactivé la version 3.8.17 du moteur RabbitMQ en raison d'un problème lié aux files d'attente utilisant le protocole TTL (par message). time-to-live Nous vous recommandons d'utiliser la version 3.8.11.
29 juillet 2021	Amazon MQ for RabbitMQ prend désormais en charge RabbitMQ version 3.8.17. Pour plus d'informations sur les correctifs et fonctionnalités contenus dans cette mise à jour, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.8.17 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ • Gestion des versions du moteur Amazon MQ for RabbitMQ

Date	Mise à jour de la documentation
16 juillet 2021	Vous pouvez désormais ajuster la fenêtre de maintenance d'un courtier Amazon MQ à l'aide de AWS Management Console AWS CLI, ou de l'API Amazon MQ. Pour en savoir plus sur les fenêtres de maintenance de l’agent, consultez la documentation suivante. • Planification de la fenêtre de maintenance pour un courtier Amazon MQ
6 juillet 2021	Amazon MQ for RabbitMQ introduit la prise en charge du type d’échange de hachage cohérent. Le hachage cohérent échange les messages d’acheminement vers les files d’attente en fonction d’une valeur de hachage calculée à partir de la clé de routage d’un message. Pour plus d’informations, consultez les ressources suivantes : • Plugin d'échange de hachage cohérent • Type d'échange de hachage cohérent RabbitMQ sur le référentiel RabbitMQ GitHub
7 juin 2021	Amazon MQ prend désormais en charge ActiveMQ 5.16.2, une nouvelle version majeure du moteur. Pour plus d’informations, consultez les ressources suivantes : • Notes de mise à jour d’ActiveMQ • Gestion des versions du moteur Amazon MQ for ActiveMQ • Mise à niveau d'une version du moteur d'agent Amazon MQ • Utilisation des fichiers de configuration XML Spring
26 mai 2021	Amazon MQ for RabbitMQ est désormais disponible dans les régions de Chine (Beijing) et de Chine (Ningxia). Pour de plus amples informations sur les régions disponibles, veuillez consultez Régions et points de terminaison AWS.

Date	Mise à jour de la documentation
18 mai 2021	Amazon MQ for RabbitMQ implémente les valeurs par défaut de l'agent. Lorsque vous créez un agent pour la première fois, Amazon MQ crée un ensemble de politiques de l'agent et de limites de vhost en fonction du type d'instance et du mode de déploiement que vous choisissez, afin d'optimiser les performances de l'agent. Pour plus d'informations, consultez les rubriques suivantes : https://docs.aws.amazon.com//amazon-mq/latest/developer-guide/rabbitmq-defaults.html
5 mai 2021	Amazon MQ prend désormais en charge ActiveMQ 5.15.15. Pour plus d'informations, consultez les ressources suivantes : • Notes de mise à jour d'ActiveMQ 5.15.15 • Gestion des versions du moteur Amazon MQ for ActiveMQ • Utilisation des fichiers de configuration XML Spring
5 mai 2021	Amazon MQ a commencé à suivre les modifications apportées aux politiques AWS gérées. Pour plus d'informations, consultez les ressources suivantes : • the section called "AWS politiques gérées"
14 avril 2021	Amazon MQ est désormais disponible dans les régions de Chine (Beijing) et de Chine (Ningxia). Pour de plus amples informations sur les régions disponibles, veuillez consultez Régions et points de terminaison AWS.
7 avril 2021	Amazon MQ prend désormais en charge RabbitMQ 3.8.11. Pour plus d'informations sur les correctifs et fonctionnalités contenus dans cette mise à jour, consultez la documentation suivante : • Notes de mise à jour de RabbitMQ 3.8.11 sur le référentiel du serveur RabbitMQ GitHub • Journal des modifications RabbitMQ • Gestion des versions du moteur Amazon MQ for RabbitMQ

Date	Mise à jour de la documentation
1 avril 2021	Amazon MQ est maintenant disponible dans la région d'Asie-Pacifique (Osaka). Pour plus d'information sur les régions disponibles , consultez Régions et points de terminaison Amazon MQ.
21 décembre 2020	Amazon MQ prend désormais en charge ActiveMQ 5.15.14. Pour plus d'informations, consultez les ressources suivantes : • Notes de mise à jour d'ActiveMQ • Gestion des versions du moteur Amazon MQ for ActiveMQ • Utilisation des fichiers de configuration XML Spring • **Important** En raison d'un problème connu avec Apache ActiveMQ dans cette version, le nouveau bouton Pause Queue (Mettre en pause la file d'attente) de la console Web ActiveMQ ne peut pas être utilisé avec Amazon MQ pour les agents ActiveMQ. Pour de plus amples informations sur le problème, veuillez consulter AMQ-8104.

Date	Mise à jour de la documentation
4 novembre 2020	Amazon MQ prend désormais en charge RabbitMQ, un agent de messages open source populaire. Cela vous permet de migrer vos courtiers de messages RabbitMQ existants AWS sans avoir à réécrire le code. Amazon MQ for RabbitMQ gère à la fois les agents de messages individue ls et en cluster et gère des tâches telles que le provisionnement de l'infrast ructure, la configuration de l'agent et la mise à jour du logiciel. • Amazon MQ prend en charge RabbitMQ 3.8.6. Pour de plus amples informations sur les versions du moteur pris en charge, veuillez consulter the section called "Gestion des versions". • L'offre gratuite AWS inclut jusqu'à 750 heures d'utilisation d'un agent `mq.t3.micro` à une instance et jusqu'à 20 Go de stockage par mois pendant un an. Pour plus d'informations sur les types d'instance pris en charge, consultez Broker instance types. • Avec Amazon MQ for RabbitMQ, vous pouvez accéder à vos agents en utilisant AMQP 0-9-1, et utiliser n'importe quelle langue prise en charge par les bibliothèques client RabbitMQ. Pour plus d'informations sur les protocoles et les suites de chiffrement pris en charge, consultez the section called "Protocoles Amazon MQ for RabbitMQ". • Amazon MQ for RabbitMQ est disponible dans toutes les régions où Amazon MQ est actuellement disponible. Pour en savoir plus sur toutes les régions disponibles, consultez le tableau des régions AWS. Pour commencer à utiliser Amazon MQ, créez un agent et connectez une application axée sur JVM à votre agent RabbitMQ, consultez Mise en route : création et connexion à un courtier RabbitMQ.
22 octobre 2020	Amazon MQ prend en charge ActiveMQ 5.15.13. Pour plus d'informations, consultez les ressources suivantes : • Notes de mise à jour d'ActiveMQ • Gestion des versions du moteur Amazon MQ for ActiveMQ • Utilisation des fichiers de configuration XML Spring

Date	Mise à jour de la documentation
30 septembre 2020	Amazon MQ est désormais disponible dans la région de l'UE (Milan). Pour plus d'information sur les régions et les points de terminaison disponibles, consultez Régions et points de terminaison Amazon MQ.
27 juillet 2020	Vous pouvez authentifier les utilisateurs Amazon MQ à l'aide des informations d'identification stockées dans votre Active Directory ou un autre serveur LDAP. Vous pouvez également ajouter, supprimer et modifier des utilisateurs Amazon MQ et attribuer des autorisations aux rubriques et aux files d'attente. Pour de plus amples informations, veuillez consulter Intégrer LDAP avec ActiveMQ.
17 juillet 2020	Amazon MQ prend désormais en charge le type d'instance `mq.t3.micro`. Pour de plus amples informations, veuillez consulter Broker instance types.
30 juin 2020	Amazon MQ prend en charge ActiveMQ 5.15.12. Pour plus d'informations, consultez les ressources suivantes : • Notes de mise à jour d'ActiveMQ • Gestion des versions du moteur Amazon MQ for ActiveMQ • Utilisation des fichiers de configuration XML Spring

Date	Mise à jour de la documentation
30 avril 2020	Amazon MQ prend en charge un nouvel élément de collection enfant, `systemUsage` , sur l'élément `broker`. Pour de plus amples informations, veuillez consulter systemUsage. Amazon MQ prend également en charge trois nouveaux attributs sur l'élément enfant `kahaDB`. • `journalDiskSyncInterval` - Intervalle (ms) indiquant quand effectuer une synchronisation de disque si `journalDiskSyncStrategy=periodic` . • `journalDiskSyncStrategy` - configure la stratégie de synchronisation du disque. • `preallocationStrategy` - configure la façon dont l'agent va essayer de préallouer les fichiers journaux lorsqu'un nouveau fichier journal est nécessaire. Pour de plus amples informations, veuillez consulter Attributes.
3 mars 2020	Amazon MQ prend en charge deux nouvelles métriques CloudWatch • `TempPercentUsage` - Pourcentage de stockage temporaire disponible utilisé par les messages non persistants. • `JobSchedulerStorePercentUsage` - Pourcentage d'espace disque utilisé par le magasin du planificateur de tâches. Pour de plus amples informations, veuillez consulter Monitoring and logging Amazon MQ brokers.
4 février 2020	Amazon MQ est disponible dans les régions d'Asie-Pacifique (Hong Kong) et du Moyen-Orient (Bahreïn). Pour de plus amples informations sur les régions disponibles, veuillez consultez Régions et points de terminaison AWS.

Date	Mise à jour de la documentation
22 janvier 2020	Amazon MQ prend en charge ActiveMQ 5.15.10. Pour plus d'informations, consultez les ressources suivantes : • Notes de mise à jour d'ActiveMQ • Gestion des versions du moteur Amazon MQ for ActiveMQ • Utilisation des fichiers de configuration XML Spring
19 décembre 2019	Amazon MQ est disponible dans les régions d'UE (Stockholm) et d'Amériqu e du Sud (São Paulo). Pour de plus amples informations sur les régions disponibles, veuillez consultez Régions et points de terminaison AWS.

Date	Mise à jour de la documentation
16 décembre 2019	Amazon MQ prend en charge la création d'agents optimisés pour le débit à l'aide d'Amazon Elastic Block Store (EBS), au lieu d'Amazon Elastic File System (Amazon EFS) par défaut, pour le stockage d'agents. Pour tirer parti d'une grande durabilité et d'une réplication sur plusieurs zones de disponibilité, utilisez Amazon EFS. Pour profiter d'une faible latence et d'un débit élevé, utilisez Amazon EBS. **Important** • Vous pouvez utiliser Amazon EBS uniquement avec la gamme de type d'instance d'agent `mq.m5`. • Bien que vous puissiez modifier le type d'instance de l'agent, vous ne pouvez pas modifier le type de stockage de l'agent après avoir créé l'agent. • Amazon EBS réplique les données dans une seule zone de disponibilité et ne prend pas en charge le mode de déploiement actif/en veille ActiveMQ. Pour plus d'informations, consultez les ressources suivantes : • Storage • Choisir le type de stockage d'agent adéquat pour un débit optimal • La propriété `storageType` de la ressource `broker-instance-options` dans la référence des API REST Amazon MQ • Les mesures `BurstBalance`, `VolumeReadOps` et `VolumeWriteOps` dans la section Monitoring and logging Amazon MQ brokers.
18 octobre 2019	Deux CloudWatch statistiques Amazon sont disponibles : `TotalEnqueueCount` et `TotalDequeueCount`. Pour plus d'informations, voir Monitoring and logging Amazon MQ brokers

Date	Mise à jour de la documentation
11 octobre 2019	Amazon MQ prend désormais en charge les points de terminaison conformes à la norme FIPS (Federal Information Processing Standard) 140-2 dans les régions commerciales aux États-Unis. Pour plus d'informations, consultez les ressources suivantes : • Norme fédérale de traitement de l'information (FIPS) 140-2 • Régions et points de terminaison Amazon MQ
30 septembre 2019	Amazon MQ offre désormais la possibilité de mettre à l'échelle vos agents en modifiant le type d'instance hôte. Pour plus d'informations, consultez la propriété `hostInstanceType` de `UpdateBrokerInput` et la propriété `pendingHostInstanceType` de `DescribeBrokerOutput`.
30 août 2019	Vous pouvez désormais mettre à jour les groupes de sécurité associés à un agent, à la fois dans la console et avec `UpdateBrokerInput`.
22 juillet 2019	Amazon MQ s'intègre à AWS Key Management Service (KMS) pour proposer un chiffrement côté serveur. Vous pouvez désormais sélectionner votre propre clé CMK gérée par le client ou utiliser une clé KMS AWS gérée dans votre AWS KMS compte. Pour de plus amples informations, veuillez consulter Chiffrement au repos. Amazon MQ prend en charge l'utilisation AWS KMS des clés des manières suivantes. • AWS clé KMS détenue : la clé appartient à Amazon MQ et ne figure pas dans votre compte. • AWS clé KMS AWS gérée : la clé KMS gérée (aws/mq) est une clé KMS de votre compte créée, gérée et utilisée en votre nom par Amazon MQ. • Sélectionnez une clé CMK gérée par le client existante — Les clients gérés CMKs sont créés et gérés par vous dans AWS Key Management Service (KMS).

Date	Mise à jour de la documentation
19 juin 2019	Amazon MQ est disponible dans les régions de l'UE (Paris) et de l'Asie-Pacifique (Mumbai). Pour de plus amples informations sur les régions disponibles, veuillez consultez Régions et points de terminaison AWS.
12 juin 2019	Amazon MQ est désormais disponible dans la région Canada (Centre) Pour de plus amples informations sur les régions disponibles, veuillez consultez Régions et points de terminaison AWS.
3 juin 2019	Deux nouvelles CloudWatch statistiques Amazon sont disponibles : `EstablishedConnectionsCount` et `InactiveDurableSubscribers`. Pour plus d'informations, consultez les ressources suivantes : • Monitoring and logging Amazon MQ brokers • Monitoring and logging Amazon MQ brokers
10 mai 2019	Le stockage de données pour les nouveaux types d'instance `mq.t2.micro` est limité à 20 Go. Pour plus d'informations, consultez les ressources suivantes : • the section called "Stockage des données" • Broker instance types
29 avril 2019	Vous pouvez désormais utiliser des stratégies axées sur des balises et des autorisations au niveau des ressources. Pour plus d'informations, consultez les ressources suivantes : • Fonctionnement d'Amazon MQ avec IAM • Autorisations au niveau des ressources pour les actions d'API Amazon MQ
16 avril 2019	Vous pouvez désormais récupérer des informations sur le moteur d'agent et les options d'instance d'agent à l'aide de l'API REST. Pour plus d'informations, consultez les ressources suivantes : • Options d'instance d'agent • Types de moteur d'agent

Date	Mise à jour de la documentation
8 avril 2019	Amazon MQ prend en charge ActiveMQ 5.15.9. Pour plus d'informations, consultez les ressources suivantes : • Notes de mise à jour d'ActiveMQ • Gestion des versions du moteur Amazon MQ for ActiveMQ • Utilisation des fichiers de configuration XML Spring
4 mars 2019	Amélioration de la documentation sur la configuration du basculement dynamique et du rééquilibrage de clients pour un réseau d'agents. Activez le basculement dynamique en configurant `transportConnectors` ainsi que des options de configuration `networkConnectors` . Pour plus d'informations, consultez les ressources suivantes : • Basculement dynamique avec des connecteurs de transport • Réseau de courtiers Amazon MQ • Amazon MQ Broker Configuration Parameters
27 février 2019	Amazon MQ est disponible dans la région UE (Londres), en plus des régions suivantes : • Asie-Pacifique (Singapour) • USA Est (Ohio) • USA Est (Virginie du Nord) • USA Ouest (Californie du Nord) • USA Ouest (Oregon) • Asie Pacifique (Tokyo) • Asie-Pacifique (Séoul) • Asie-Pacifique (Sydney) • Europe (Francfort) • Europe (Irlande)
24 janvier 2019	La configuration par défaut inclut désormais une stratégie pour purger les destinations inactives.

Date	Mise à jour de la documentation
le 17 janvier 2019	Les types d'instance `mq.t2.micro` Amazon MQ prennent désormais en charge seulement 100 connexions par protocole de niveau filaire. Pour plus d'informations, veuillez consulter Quotas in Amazon MQ.
19 décembre 2018	Vous pouvez configurer une série d'agents Amazon MQ dans un réseau d'agents. Pour plus d'informations, consultez les sections suivantes : • Réseau de courtiers Amazon MQ • Creating and Configuring a Network of Brokers • Correctement configurer votre réseau d'agents • `networkConnector` • `networkConnectionStartAsynchrone`

11 décembre 2018	Amazon MQ prend en charge ActiveMQ 5.15.8, 5.15.6 et 5.15.0.
	• « Resolved bugs and improvements » (Bugs résolus et améliorations) dans la documentation ActiveMQ :
	  • Notes de mise à jour d'ActiveMQ 5.15.8
	  • Notes de mise à jour d'ActiveMQ 5.15.7

5 décembre 2018	AWS prend en charge le balisage des ressources pour faciliter le suivi de votre allocation des coûts. Vous pouvez baliser des ressources lorsque vous les créez, ou en consultant les détails de cette ressource. Pour plus d'informations, consultez Balisage de vos ressources.
19 novembre 2018	AWS a étendu son programme de conformité SOC pour inclure Amazon MQ en tant que service conforme au SOC.
15 octobre 2018	• Le nombre maximal de groupes par utilisateur est de 20. Pour de plus amples informations, veuillez consulter Users. • Le nombre maximal de connexions par agent et par protocole de niveau filaire est de 1 000. Pour de plus amples informations, veuillez consulter Agents.

Date	Mise à jour de la documentation
2 octobre 2018	AWS a étendu son programme de conformité à la loi HIPAA pour inclure Amazon MQ en tant que service éligible à la loi HIPAA.

le 27 septembre 2018	Amazon MQ prend en charge ActiveMQ 5.15.6, en plus de 5.15.0. Pour plus d'informations, consultez les ressources suivantes :
	• Mise en route : création et connexion à un courtier ActiveMQ
	• Resolved bugs and improvements dans la documentation ActiveMQ :
	  • Notes de mise à jour d'ActiveMQ 5.15.6
	  • Notes de mise à jour d'ActiveMQ 5.15.5
	  • Notes de mise à jour d'ActiveMQ 5.15.4
	  • Notes de mise à jour d'ActiveMQ 5.15.3
	  • Notes de mise à jour d'ActiveMQ 5.15.2
	  • Notes de mise à jour d'ActiveMQ 5.15.1
	• Client ActiveMQ 5.15.6

31 août 2018	• Les mesures suivantes sont disponibles :
	  • `CurrentConnectionsCount`
	  • `TotalConsumerCount`
	  • `TotalProducerCount`
	  Pour plus d'informations, consultez la section Monitoring and logging Amazon MQ brokers.
	• L'adresse IP de l'agent est affichée sur la page Details (Informations).
	  Note
	  Pour les agents avec l'accessibilité publique désactivée, l'adresse IP interne s'affiche.

Date	Mise à jour de la documentation
30 août 2018	Amazon MQ est disponible dans la région d'Asie-Pacifique (Singapour), en plus des régions suivantes : • USA Est (Ohio) • USA Est (Virginie du Nord) • USA Ouest (Californie du Nord) • USA Ouest (Oregon) • Asie Pacifique (Tokyo) • Asie-Pacifique (Séoul) • Asie-Pacifique (Sydney) • Europe (Francfort) • Europe (Irlande)
30 juillet 2018	Vous pouvez configurer Amazon MQ pour publier des journaux généraux et d'audit sur Amazon CloudWatch Logs. Pour de plus amples informations, veuillez consulter Monitoring and logging Amazon MQ brokers.
25 juillet 2018	Amazon MQ est disponible dans les régions d'Asie-Pacifique (Tokyo) et d'Asie-Pacifique (Séoul), en plus des régions suivantes : • USA Est (Ohio) • USA Est (Virginie du Nord) • USA Ouest (Californie du Nord) • US West (Oregon) • Asie-Pacifique (Sydney) • Europe (Francfort) • Europe (Irlande)
19 juillet 2018	Vous pouvez l'utiliser AWS CloudTrail pour enregistrer les appels d'API Amazon MQ. Pour de plus amples informations, veuillez consulter Logging Amazon MQ API calls using CloudTrail.

Date	Mise à jour de la documentation
29 juin 2018	En plus de `mq.t2.micro` et `mq.m4.large`, les types d'instance d'agent suivants sont disponibles pour le développement, test et production de charges de travail nécessitant un débit élevé : • `mq.m5.large` • `mq.m5.xlarge` • `mq.m5.2xlarge` • `mq.m5.4xlarge` Pour de plus amples informations, veuillez consulter Broker instance types.
27 juin 2018	Amazon MQ est disponible dans la région d'USA Ouest (Californie du Nord), en plus des régions suivantes : • USA Est (Ohio) • USA Est (Virginie du Nord) • USA Ouest (Oregon) • Asie-Pacifique (Sydney) • Europe (Francfort) • Europe (Irlande)

Date	Mise à jour de la documentation

14 juin 2018	• Vous pouvez utiliser la `AWS::Amazon MQ::Broker` AWS CloudFormation ressource pour effectuer les actions suivantes :
	  • Créer un agent.
	  • Ajouter des modifications de configuration ou modifier les utilisateurs pour l'agent précisé.
	  • Renvoyer des informations sur l'agent précisé.
	  • Supprimer l'agent précisé.
	
	  Note
	  Lorsque vous modifiez une propriété du type de propriété Amazon MQ Broker ConfigurationId ou Amazon MQ Broker User, le courtier est redémarré immédiatement.
	• Vous pouvez utiliser la `AWS::Amazon MQ::Configuration` AWS CloudFormation ressource pour effectuer les actions suivantes :
	  • Créer une configuration.
	  • Mettre à jour la configuration précisée.
	  • Renvoyer des informations sur la configuration précisée.
	
	  Note
	  Vous pouvez l'utiliser CloudFormation pour modifier, mais pas supprimer, une configuration Amazon MQ.

7 juin 2018	La console Amazon MQ prend en charge l'allemand, le portugais brésilien, l'espagnol, l'italien et le chinois traditionnel.
17 mai 2018	La limite de nombre d'utilisateurs par agent est de 250. Pour de plus amples informations, veuillez consulter Users.
13 mars 2018	La création d'un agent prend environ 15 minutes. Pour en savoir plus, consultez la page Terminer la création d'un agent.

Date	Mise à jour de la documentation
1er mars 2018	• Vous pouvez configurer la répartition et le stockage simultanés pour Apache KahaDB à l'aide de l'attribut `concurrentStoreAndDispatchQueues`. • La `CpuCreditBalance` CloudWatch métrique > est disponible pour le type d'instance de `mq.t2.micro` courtier.
10 janvier 2018	Les modifications suivantes affectent la console Amazon MQ : • Dans la liste d'agents, la colonne Creation (Création) est masquée par défaut. Pour personnaliser la taille de page et les colonnes, choisissez  . • Sur la ***`MyBroker`***page, dans la section Connexion s, choisissez le nom de votre groupe de sécurité ou  ouvrez la console EC2 (au lieu de la console VPC). La console EC2 permet une configuration plus intuitive des règles entrantes et sortantes. Pour en savoir plus, consultez la section Connecting a Java application to your broker mise à jour.
9 janvier 2018	• L'autorisation pour l'ID d'opération REST `UpdateBroker` est répertoriée correctement comme `mq:UpdateBroker` dans la console IAM. • L'autorisation erronée `mq:DescribeEngine` est supprimée de la console IAM.

Date	Mise à jour de la documentation

28 novembre 2017	Il s'agit de la première version d'Amazon MQ et du Guide du développeur Amazon MQ.
	
	• Amazon MQ est disponible dans les régions suivantes :
	  • USA Est (Ohio)
	  • USA Est (Virginie du Nord)
	  • USA Ouest (Oregon)
	  • Asie-Pacifique (Sydney)
	  • Europe (Francfort)
	  • Europe (Irlande)
	
	  L'utilisation du `mq.t2.micro` type d'instance est soumise aux crédits UC et performance de base, avec la possibilité de débordement au-dessus du niveau de base (pour plus d'informations, consultez la `CpuCreditBalance` métrique). Si votre application nécessite des performances fixes, envisagez d'utiliser un type d'instance `mq.m5.large` .
	• Vous pouvez créer des agents `mq.m4.large` et `mq.t2.micro` .
	
	  L'utilisation du `mq.t2.micro` type d'instance est soumise aux crédits UC et performance de base, avec la possibilité de débordement au-dessus du niveau de base (pour plus d'informations, consultez la `CpuCreditBalance` métrique). Si votre application nécessite des performances fixes, envisagez d'utiliser un `mq.m5.large` type d'instance.
	• Vous pouvez utiliser le moteur d'agent ActiveMQ 5.15.0.
	• Vous pouvez également créer et gérer des courtiers par programmation à l'aide de l'API REST Amazon MQ et. AWS SDKs
	• Vous pouvez accéder à vos agents via tout langage de programmation pris en charge par ActiveMQ et en activant explicitement TLS pour les protocoles suivants :
	  • AMQP
	  • MQTT
	  • MQTT terminé WebSocket
	  • OpenWire

Date	Mise à jour de la documentation
	• STOMP • STOMP over WebSocket • Vous pouvez vous connecter à des agents ActiveMQ à l'aide de différents clients ActiveMQ. Nous vous recommandons d'utiliser le client ActiveMQ. Pour de plus amples informations, veuillez consulter Connecting a Java application to your broker. • Votre agent peut envoyer et recevoir des messages de n'importe quelle taille.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.