aws

Guide du développeur

Amazon DocumentDB

Amazon DocumentDB: Guide du développeur

Table of Contents

Qu'est-ce qu'Amazon DocumentDB (avec compatibilité avec MongoDB)

Amazon DocumentDB (compatible avec MongoDB) est un service de base de données rapide, fiable et entièrement géré. Amazon DocumentDB facilite la configuration, l'exploitation et le dimensionnement de bases de données compatibles avec MongoDB dans le cloud. Avec Amazon DocumentDB, vous pouvez exécuter le même code d'application et utiliser les mêmes pilotes et outils que ceux que vous utilisez avec MongoDB.

Avant d'utiliser Amazon DocumentDB, vous devez consulter les concepts et fonctionnalités décrits dans. Comment ça marche Ensuite, complétez les étapes de Guide de démarrage.

Rubriques

Présentation d'Amazon DocumentDB

Voici quelques fonctionnalités de haut niveau d'Amazon DocumentDB :

- Amazon DocumentDB prend en charge deux types de clusters : les clusters basés sur des instances et les clusters élastiques. Les clusters élastiques supportent des charges de travail comportant des millions de lectures/écritures par seconde et une capacité de stockage de plusieurs pétaoctets. Pour plus d'informations sur les clusters élastiques, consultezUtilisation des clusters élastiques Amazon DocumentDB. Le contenu ci-dessous fait référence aux clusters basés sur des instances Amazon DocumentDB.

- Amazon DocumentDB augmente automatiquement la taille de votre volume de stockage à mesure que vos besoins de stockage de base de données augmentent. Votre volume de stockage augmente par paliers de 10 Go, jusqu'à un maximum de 128 TiB. Vous n'avez pas besoin de prévoir d'espace de stockage supplémentaire pour maîtriser la croissance future de votre cluster.
- Avec Amazon DocumentDB, vous pouvez augmenter le débit de lecture pour prendre en charge de gros volumes de demandes d'applications en créant jusqu'à 15 instances de réplication. Les répliques Amazon DocumentDB partagent le même stockage sous-jacent, ce qui réduit les coûts et évite d'avoir à effectuer des écritures sur les nœuds de réplication. Cette fonctionnalité libère davantage de puissance de traitement pour traiter les demandes de lecture et réduit le délai de réplication, souvent jusqu'à quelques millisecondes à un chiffre. Vous pouvez ajouter des répliques en quelques minutes, quelle que soit la taille du volume de stockage. Amazon DocumentDB fournit également un point de terminaison pour le lecteur, qui permet à l'application de se connecter sans avoir à suivre les répliques au fur et à mesure de leur ajout ou de leur suppression.
- Amazon DocumentDB vous permet d'augmenter ou de diminuer les ressources de calcul et de mémoire de chacune de vos instances. Les opérations de mise à l'échelle du calcul sont normalement réalisées en quelques minutes.
- Amazon DocumentDB s'exécute dans Amazon Virtual Private Cloud (Amazon VPC), ce qui vous permet d'isoler votre base de données dans votre propre réseau virtuel. Vous pouvez également configurer vos paramètres de pare-feu pour contrôler l'accès réseau à votre cluster.
- Amazon DocumentDB surveille en permanence l'état de santé de votre cluster. En cas de défaillance d'une instance, Amazon DocumentDB redémarre automatiquement l'instance et les processus associés. Amazon DocumentDB ne nécessite pas de relecture des journaux de restauration des bases de données en cas de panne, ce qui réduit considérablement les temps de redémarrage. Amazon DocumentDB isole également le cache de base de données du processus de base de données, ce qui permet au cache de survivre au redémarrage d'une instance.
- En cas de défaillance d'une instance, Amazon DocumentDB automatise le basculement vers l'une des 15 répliques Amazon DocumentDB que vous créez dans d'autres zones de disponibilité. Si aucune réplique n'a été mise en service et qu'une défaillance survient, Amazon DocumentDB essaie de créer automatiquement une nouvelle instance Amazon DocumentDB.
- La fonctionnalité de sauvegarde d'Amazon DocumentDB permet la point-in-time restauration de votre cluster. Cette fonction vous permet de restaurer votre cluster d'une seconde au cours de la période de rétention, jusqu'aux 5 dernières minutes. Vous pouvez configurer votre période de rétention des sauvegardes automatique de 35 jours maximum. Les sauvegardes automatisées sont stockées dans Amazon Simple Storage Service (Amazon S3), conçu pour une durabilité de

99,999999999 %. Les sauvegardes Amazon DocumentDB sont automatiques, incrémentielles et continues, et elles n'ont aucun impact sur les performances de votre cluster.

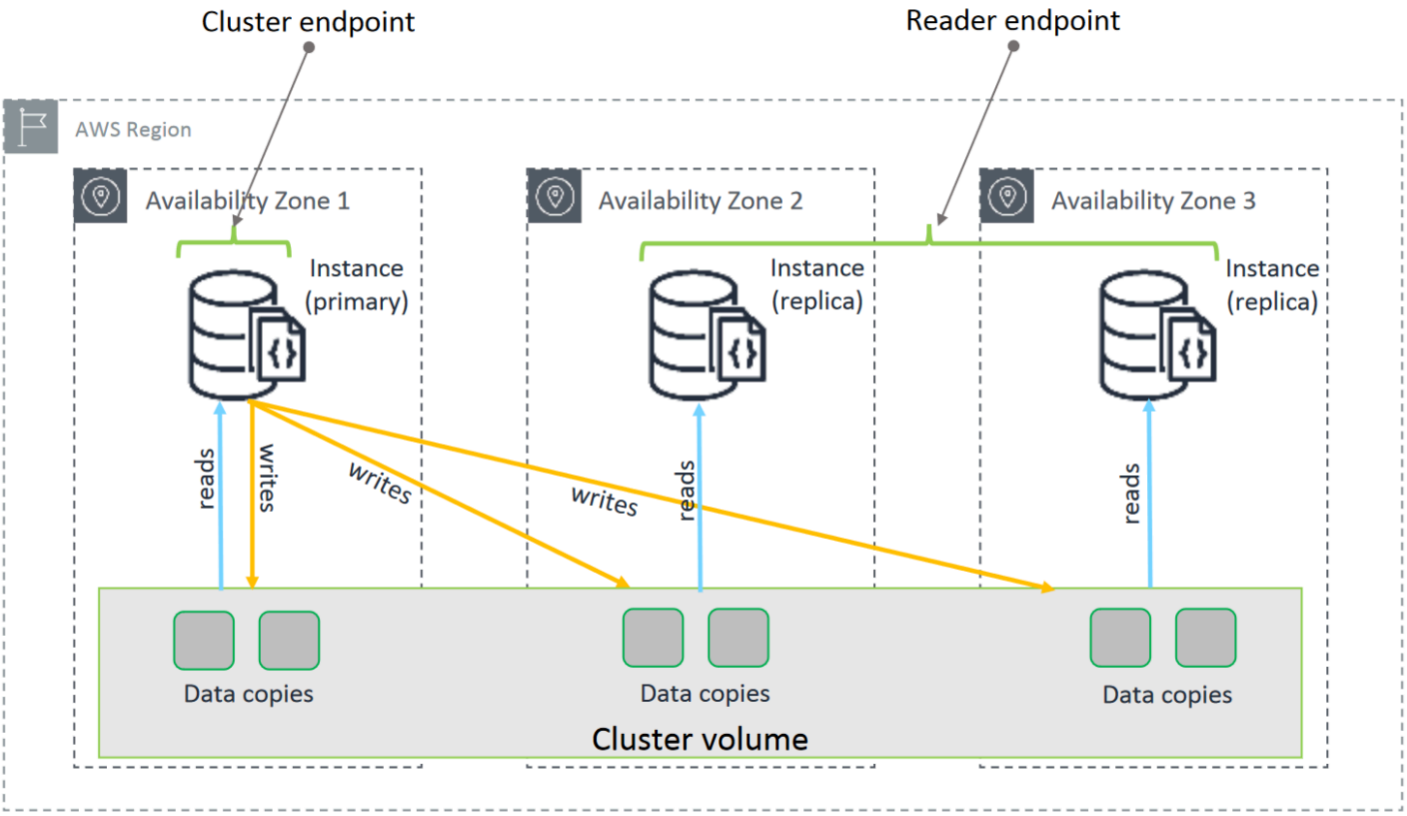
- Avec Amazon DocumentDB, vous pouvez chiffrer vos bases de données à l'aide de clés que vous créez et contrôlez via AWS Key Management Service ().AWS KMS Sur un cluster de base de données exécuté avec le chiffrement Amazon DocumentDB, les données stockées au repos dans le stockage sous-jacent sont chiffrées. Les sauvegardes automatisées, les instantanés et les réplicas dans le même cluster sont également chiffrés.

Si vous débutez dans le AWS domaine des services, consultez les ressources suivantes pour en savoir plus :

- AWS propose des services informatiques, de bases de données, de stockage, d'analyse et d'autres fonctionnalités. Pour un aperçu de tous les AWS services, consultez Cloud Computing with Amazon Web Services.
- AWS fournit un certain nombre de services de base de données. Pour savoir quel service est le mieux adapté à votre environnement, voir Bases de données sur AWS.

Clusters

Un cluster se compose de 0 à 16 instances et d'un volume de stockage de cluster qui gère les données de ces instances. Toutes les écriture sont effectuées à travers l'instance principale. Toutes les instances (principale et réplicas) prennent en charge les lectures. Les données du cluster sont stockées dans le volume de cluster avec des copies dans trois zones de disponibilité différentes.



Les clusters basés sur des instances Amazon DocumentDB 5.0 prennent en charge deux configurations de stockage pour un cluster de base de données : Amazon DocumentDB standard et Amazon DocumentDB optimisé pour les E/S. Pour plus d'informations, consultez Configurations de stockage en cluster Amazon DocumentDB.

instances

Une instance Amazon DocumentDB est un environnement de base de données isolé dans le cloud. Une instance peut comporter plusieurs bases de données créées par l'utilisateur. Vous pouvez créer et modifier une instance à l'aide du AWS Management Console ou du AWS CLI.

La capacité de calcul et de mémoire d'une instance est déterminée par sa classe d'instance. Vous pouvez sélectionner l'instance qui correspond le mieux à vos besoins. Si vos besoins évoluent au fil du temps, vous pouvez choisir une autre classe d'instance. Pour connaître les spécifications de classes, veuillez consulter Spécifications de la classe d'instance.

Les instances Amazon DocumentDB s'exécutent uniquement dans l'environnement Amazon VPC. Amazon VPC vous permet de contrôler votre environnement réseau virtuel : vous pouvez choisir

votre propre plage d'adresses IP, créer des sous-réseaux et configurer des listes de routage et de contrôle d'accès (ACL).

Avant de créer des instances Amazon DocumentDB, vous devez créer un cluster contenant les instances.

Toutes les classes d'instances ne sont pas prises en charge dans toutes les régions. Le tableau suivant spécifie les classes d'instances prises en charge dans chaque région.

Classes d'instances prises en charge par région

Région	R6G	R5	R4	T4G	T3
USA Est (Ohio)	Pris en charge	Pris en charge	Pris en charge	Pris en charge	Pris en charge
USA Est (Virginie du Nord)	Pris en charge	Pris en charge	Pris en charge	Pris en charge	Pris en charge
USA Ouest (Oregon)	Pris en charge	Pris en charge	Pris en charge	Pris en charge	Pris en charge
Amérique du Sud (São Paulo)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Asie-Pacifique (Hong Kong)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Asie-Pacifique (Hyderabad)		Pris en charge			Pris en charge
Asie-Pacifique (Mumbai)	Pris en charge	Pris en charge		Pris en charge	Pris en charge

Région	R6G	R5	R4	T4G	T3
Asie-Pacifique (Séoul)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Asie-Pacifique (Sydney)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Asie-Pacifique (Singapour)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Asie-Pacifique (Tokyo)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Canada (Centre)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Europe (Francfort)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Europe (Irlande)	Pris en charge	Pris en charge	Pris en charge	Pris en charge	Pris en charge
Europe (Londres)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Europe (Milan)	Pris en charge	Pris en charge		Pris en charge	Pris en charge

Région	R6G	R5	R4	T4G	T3
Europe (Paris)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Moyen-Orient (EAU)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Région Chine (Beijing)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Chine (Ningxia)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
AWS GovCloud (US-Ouest)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
AWS GovCloud (USA Est)	Pris en charge	Pris en charge		Pris en charge	Pris en charge

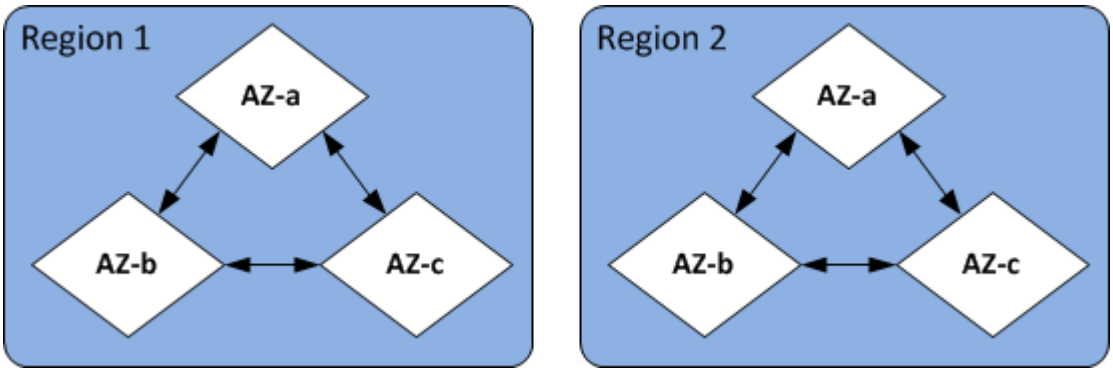
Régions et zones de disponibilité

Les régions et les zones de disponibilité définissent les emplacements physiques de votre cluster et de vos instances.

Régions

AWS Les ressources de cloud computing sont hébergées dans des centres de données hautement disponibles dans différentes régions du monde (par exemple, en Amérique du Nord, en Europe ou en Asie). Chaque emplacement de centre de données est appelé région.

Chaque AWS région est conçue pour être complètement isolée des autres AWS régions. Chaque région contient plusieurs zones de disponibilité. En lançant vos nœuds dans différentes zones de disponibilité, vous pouvez obtenir la plus grande tolérance aux pannes possible. Le schéma suivant montre une vue d'ensemble du fonctionnement des AWS régions et des zones de disponibilité.



Zones de disponibilité

Chaque AWS région contient plusieurs emplacements distincts appelés zones de disponibilité. Chaque zone de disponibilité est conçue pour être isolée des pannes dans les autres zones de disponibilité et pour fournir une connectivité réseau peu coûteuse et à faible latence vers d'autres zones de disponibilité de la même région. En lançant des instances pour un cluster donné dans plusieurs zones de disponibilité, vous pouvez protéger vos applications contre l'événement improbable de l’échec d'une zone de disponibilité.

L'architecture Amazon DocumentDB sépare le stockage et le calcul. Pour la couche de stockage, Amazon DocumentDB réplique six copies de vos données dans trois AWS zones de disponibilité. Par exemple, si vous lancez un cluster Amazon DocumentDB dans une région qui ne prend en charge que deux zones de disponibilité, votre stockage de données sera répliqué de six manières sur trois zones de disponibilité, mais vos instances de calcul ne seront disponibles que dans deux zones de disponibilité.

Le tableau suivant répertorie le nombre de zones de disponibilité que vous pouvez utiliser dans une instance donnée pour Région AWS provisionner des instances de calcul pour votre cluster.

Nom de la région	Région	Zones de disponibilité (calcul)
USA Est (Ohio)	`us-east-2`	3
USA Est (Virginie du Nord)	`us-east-1`	6

Nom de la région	Région	Zones de disponibilité (calcul)
USA Ouest (Oregon)	us-west-2	4
Amérique du Sud (São Paulo)	sa-east-1	3
Asie-Pacifique (Hong Kong)	ap-east-1	3
Asie-Pacifique (Hyderabad)	ap-south-2	3
Asie-Pacifique (Mumbai)	ap-south-1	3
Asie-Pacifique (Séoul)	ap-northeast-2	4
Asie-Pacifique (Singapour)	ap-southeast-1	3
Asie-Pacifique (Sydney)	ap-southeast-2	3
Asie-Pacifique (Tokyo)	ap-northeast-1	3
Canada (Centre)	ca-central-1	3
Région Chine (Beijing)	cn-north-1	3
Chine (Ningxia)	cn-northwest-1	3
Europe (Francfort)	eu-central-1	3
Europe (Irlande)	eu-west-1	3
Europe (Londres)	eu-west-2	3
Europe (Milan)	eu-south-1	3

Nom de la région	Région	Zones de disponibilité (calcul)
Europe (Paris)	`eu-west-3`	3
Moyen-Orient (EAU)	`me-central-1`	3
AWS GovCloud (US-Ouest)	`us-gov-west-1`	3
AWS GovCloud (USA Est)	`us-gov-east-1`	3

Tarification d'Amazon DocumentDB

Les clusters Amazon DocumentDB sont facturés sur la base des composants suivants :

- Heures d'instance (par heure) : en fonction de la classe d'instance de l'instance (par exemple,`db.r5.xlarge`). La tarification est indiquée selon une base horaire, mais les factures sont calculées à la seconde près et affichent les heures sous une forme décimale. L'utilisation d'Amazon DocumentDB est facturée par tranches d'une seconde, avec un minimum de 10 minutes. Pour plus d’informations, consultez Gestion de classes d'instance.
- Demandes d'E/S (pour 1 million de demandes par mois) : nombre total de demandes d'E/S de stockage que vous effectuez au cours d'un cycle de facturation.
- Stockage de sauvegarde (par GiB par mois) : le stockage de sauvegarde est le stockage associé aux sauvegardes de base de données automatisées et à tous les instantanés de base de données actifs que vous avez pris. Augmenter votre période de rétention des sauvegardes ou prendre des instantanés de base de données supplémentaires augmente le stockage de sauvegarde consommé par votre base de données. Le stockage de sauvegarde est mesuré en Go par mois. Le tarif par seconde ne s'applique pas. Pour plus d’informations, consultez Sauvegarde et restauration dans Amazon DocumentDB.
- Transfert de données (par Go) : transfert de données vers et depuis votre instance depuis ou vers Internet ou d'autres AWS régions.

Pour obtenir des informations détaillées, consultez la tarification d'Amazon DocumentDB.

Essai gratuit

Vous pouvez essayer Amazon DocumentDB gratuitement en utilisant l'essai gratuit d'un mois. Pour plus d'informations, consultez la section Essai gratuit dans la tarification d'Amazon DocumentDB ou consultez la FAQ relative à l'essai gratuit d'Amazon DocumentDB.

Surveillance

Il existe plusieurs façons dont vous pouvez suivre les performances et l'état d'une instance. Vous pouvez utiliser le CloudWatch service gratuit Amazon pour surveiller les performances et l'état d'une instance. Vous pouvez trouver des graphiques de performances sur la console Amazon DocumentDB. Vous pouvez vous abonner aux événements Amazon DocumentDB pour être averti lorsque des modifications sont apportées à une instance, à un instantané, à un groupe de paramètres ou à un groupe de sécurité.

Pour plus d’informations, consultez les ressources suivantes :

- Surveillance d'Amazon DocumentDB avec CloudWatch
- Journalisation des appels d'API Amazon DocumentDB à l'aide d'AWS CloudTrail

Interfaces

Vous pouvez interagir avec Amazon DocumentDB de plusieurs manières, notamment le AWS Management Console et le. AWS CLI

AWS Management Console

AWS Management Console Il s'agit d'une interface utilisateur Web simple. Vous pouvez gérer vos instances et clusters à partir de la console sans programmation requise. Pour accéder à la console Amazon DocumentDB, connectez-vous à la console Amazon DocumentDB AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.

AWS CLI

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour gérer vos clusters et instances Amazon DocumentDB. Avec une configuration minimale, vous pouvez commencer à utiliser toutes les fonctionnalités fournies par la console Amazon DocumentDB à partir de votre programme de terminal préféré.

- Pour l'installer AWS CLI, reportez-vous à la section Installation de l'interface de ligne de AWS commande.
- Pour commencer à utiliser le AWS CLI pour Amazon DocumentDB, consultez le manuel de référence de l'interface de ligne de AWS commande pour Amazon DocumentDB.

Le shell mongo

Pour vous connecter à votre cluster afin de créer, lire, mettre à jour et supprimer des documents dans vos bases de données, vous pouvez utiliser le `mongo` shell avec Amazon DocumentDB. Pour télécharger et installer le shell `mongo` 4.0, consultezÉtape 4 : Installation du shell Mongo.

Pilotes MongoDB

Pour développer et écrire des applications sur un cluster Amazon DocumentDB, vous pouvez également utiliser les pilotes MongoDB avec Amazon DocumentDB.

Quelle est la prochaine étape ?

Les sections précédentes vous ont présenté les composants d'infrastructure de base proposés par Amazon DocumentDB. Qu'allez-vous faire ensuite ? En fonction de votre situation, consultez l'une des rubriques suivantes pour commencer :

- Commencez à utiliser Amazon DocumentDB en créant un cluster et une instance à l'aide de. AWS CloudFormation Démarrage rapide avec Amazon DocumentDB AWS CloudFormation
- Commencez à utiliser Amazon DocumentDB en créant un cluster et une instance en suivant les instructions de notre. Guide de démarrage
- Commencez à utiliser Amazon DocumentDB en créant un cluster élastique en suivant les instructions fournies dans. Démarrez avec les clusters élastiques Amazon DocumentDB
- Migrez votre implémentation MongoDB vers Amazon DocumentDB en suivant les instructions fournies à l'adresse Migration vers Amazon DocumentDB

Amazon DocumentDB : comment cela fonctionne

Amazon DocumentDB (compatible avec MongoDB) est un service de base de données entièrement géré et compatible avec MongoDB. Avec Amazon DocumentDB, vous pouvez exécuter le même

code d'application et utiliser les mêmes pilotes et outils que ceux que vous utilisez avec MongoDB. Amazon DocumentDB est compatible avec MongoDB 3.6, 4.0 et 5.0.

Rubriques

Lorsque vous utilisez Amazon DocumentDB, vous commencez par créer un cluster. Un cluster de bases de données se compose de zéro ou plusieurs instances de bases de données, et d'un volume de cluster qui gère les données de ces instances. Un volume de cluster Amazon DocumentDB est un volume de stockage de base de données virtuelle qui couvre plusieurs zones de disponibilité. Chaque zone de disponibilité possède une copie des données du cluster.

Un cluster Amazon DocumentDB se compose de deux composants :

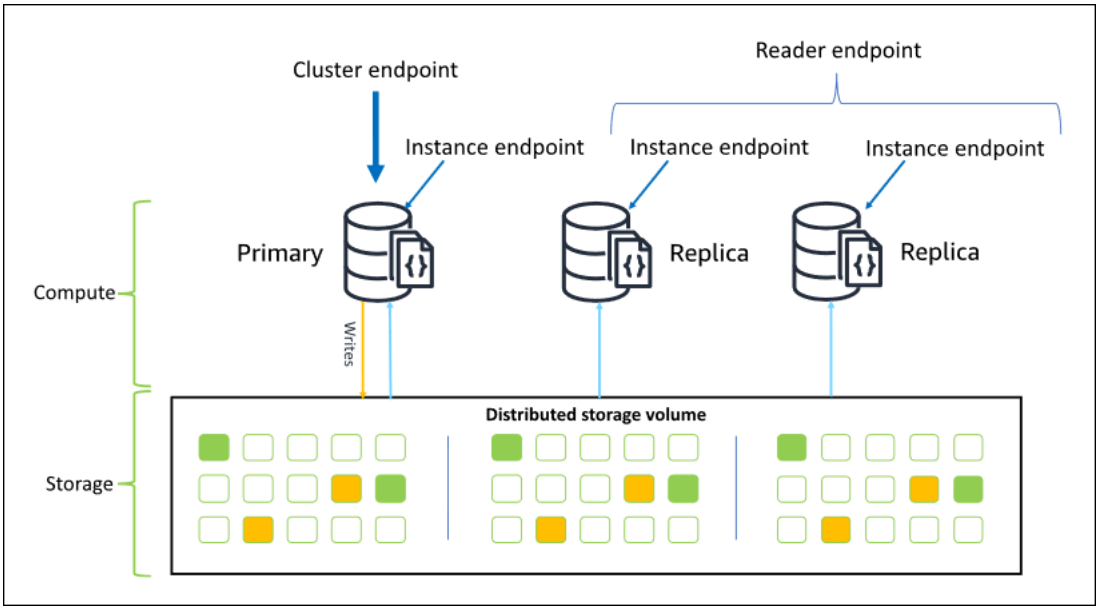
- Volume du cluster : utilise un service de stockage cloud natif pour répliquer les données de six manières sur trois zones de disponibilité, fournissant ainsi un stockage hautement durable et disponible. Un cluster Amazon DocumentDB possède exactement un volume de cluster, qui peut stocker jusqu'à 128 TiB de données.
- Instances : fournissent la puissance de traitement de la base de données, en écrivant des données sur le volume de stockage du cluster et en lisant des données à partir de celui-ci. Un cluster Amazon DocumentDB peut comporter de 0 à 16 instances.

Les instances jouent l'un de ces deux rôles :

- Instance principale : prend en charge les opérations de lecture et d'écriture et effectue toutes les modifications de données sur le volume du cluster. Chaque cluster Amazon DocumentDB possède une instance principale.

- Instance de réplication : prend en charge uniquement les opérations de lecture. Un cluster Amazon DocumentDB peut contenir jusqu'à 15 répliques en plus de l'instance principale. Le fait de posséder plusieurs réplicas vous permet de répartir les charges de travail en lecture. En outre, en plaçant les réplicas dans des zones de disponibilité distinctes, vous pouvez aussi accroître la disponibilité de votre cluster.

Le schéma suivant illustre la relation entre le volume du cluster, l'instance principale et les répliques dans un cluster Amazon DocumentDB :



Les instances de cluster n'ont pas besoin d'être de la même classe d'instance, et elles peuvent être allouées et terminées comme on le souhaite. Cette architecture vous permet de dimensionner votre capacité de calcul du cluster indépendamment de son stockage.

Lorsque votre application écrit les données sur l'instance principale, celle-ci exécute une écriture durable sur le volume de cluster. Il réplique ensuite l'état de cette écriture (et non les données) sur chaque réplique active. Les répliques Amazon DocumentDB ne participent pas au traitement des écritures. Les répliques Amazon DocumentDB sont donc avantageuses pour le dimensionnement des lectures. Les lectures à partir des répliques Amazon DocumentDB sont finalement cohérentes avec un décalage de réplication minimal, généralement moins de 100 millisecondes après que l'instance principale a écrit les données. Les lectures à partir des réplicas sont assurées d'être lues dans l'ordre dans lequel elles ont été écrits sur l'instance principale. Le retard du réplica varie en fonction de la fréquence de modification des données, et les périodes de haute activité en écriture peut augmenter le retard du réplica. Pour de plus amples informations, veuillez consulter les métriques `ReplicationLag` dans Métriques Amazon DocumentDB.

Points de terminaison Amazon DocumentDB

Amazon DocumentDB propose plusieurs options de connexion pour répondre à un large éventail de cas d'utilisation. Pour vous connecter à une instance dans un cluster Amazon DocumentDB, vous devez spécifier le point de terminaison de l'instance. Un point de terminaison est une adresse hôte et un numéro de port, séparés par un point.

Nous vous recommandons de vous connecter à votre cluster à l'aide du point de terminaison du cluster et en mode jeu de réplicas (voir Connexion à Amazon DocumentDB en tant qu'ensemble de réplicas), sauf si vous avez un cas d'utilisation spécifique pour vous connecter au point de terminaison du lecteur ou au point de terminaison d'une instance. Pour acheminer les demandes vers vos réplicas, choisissez un mode de préférence de lecture de pilote qui optimise la disponibilité en lecture tout en répondant à vos exigences de cohérence en lecture de votre application. La préférence de `secondaryPreferred` lecture active les lectures de réplica et libère l'instance principale pour effectuer plus de travail.

Les points de terminaison suivants sont disponibles à partir d'un cluster Amazon DocumentDB.

Point de terminaison de cluster

Le point de terminaison du cluster se connecte à l'instance principale actuelle de votre cluster. Des opérations de lecture et d'écriture peuvent être effectuées à l'aide du point de terminaison du cluster. Un cluster Amazon DocumentDB possède exactement un point de terminaison de cluster.

Le point de terminaison de cluster assure la prise en charge du basculement pour les connexions en lecture/écriture au cluster. En cas de défaillance de l'instance principale actuelle de votre cluster et si ce dernier possède au moins un réplica en lecture actif, le point de terminaison du cluster redirige automatiquement les demandes de connexion vers une nouvelle instance principale. Lorsque vous vous connectez à votre cluster Amazon DocumentDB, nous vous recommandons de vous connecter à votre cluster en utilisant le point de terminaison du cluster et en mode Replica Set (voirConnexion à Amazon DocumentDB en tant qu'ensemble de réplicas).

Voici un exemple de point de terminaison du cluster Amazon DocumentDB :

```
sample-cluster.cluster-123456789012.us-east-1.docdb.amazonaws.com:27017
```

L'exemple suivant présente une chaîne de connexion utilisant ce point de terminaison de cluster :

```
mongodb://username:password@sample-cluster.cluster-123456789012.us-
east-1.docdb.amazonaws.com:27017
```

Pour plus d'informations sur la recherche des points de terminaison d'un cluster, consultez Recherche des points de terminaison d'un cluster.

Point de terminaison du lecteur

Le point de terminaison du lecteur équilibre les charges des connexions en lecture seule sur tous les réplicas disponibles dans votre cluster. Un point de terminaison du lecteur de cluster fonctionnera comme le point de terminaison du cluster si vous vous connectez via le `replicaSet` mode, ce qui signifie que dans la chaîne de connexion, le paramètre du jeu de répliques est`&replicaSet=rs0`. Dans ce cas, vous pourrez effectuer des opérations d'écriture sur le primaire. Toutefois, si vous vous connectez au cluster spécifié`directConnection=true`, la tentative d'exécution d'une opération d'écriture via une connexion au point de terminaison du lecteur entraîne une erreur. Un cluster Amazon DocumentDB possède exactement un point de terminaison de lecteur.

Si le cluster ne contient qu'une instance (principale), le point de terminaison du lecteur se connecte à l'instance principale. Lorsque vous ajoutez une instance de réplique à votre cluster Amazon DocumentDB, le point de terminaison du lecteur ouvre des connexions en lecture seule vers la nouvelle réplique une fois celle-ci active.

Voici un exemple de point de terminaison de lecteur pour un cluster Amazon DocumentDB :

```
sample-cluster.cluster-ro-123456789012.us-east-1.docdb.amazonaws.com:27017
```

L'exemple suivant présente une chaîne de connexion utilisant un point de terminaison de lecteur :

```
mongodb://username:password@sample-cluster.cluster-ro-123456789012.us-
east-1.docdb.amazonaws.com:27017
```

Le point de terminaison de lecteur équilibre les charges des connexions en lecture seule, pas les demandes de lecture. Si les connexions du point de terminaison du lecteur sont plus largement utilisées que d'autres, vos demandes de lecture risquent de ne pas être correctement équilibrées entre les instances du cluster. Il est recommandé de distribuer les demandes en se connectant au point de terminaison du cluster en tant que jeu de réplicas et en utilisant l'option de préférence de lecture secondaryPreferred.

Pour plus d'informations sur la recherche des points de terminaison d'un cluster, consultez Recherche des points de terminaison d'un cluster.

Point de terminaison d'instance

Un point de terminaison d'instance se connecte à une instance spécifique dans votre cluster. Le point de terminaison d'instance pour l'instance principale actuelle peut être utilisé pour des opérations de lecture et d'écriture. Toutefois, la tentative d'effectuer des opérations d'écriture sur un point de terminaison d'instance pour un réplica en lecture se traduit par une erreur. Un cluster Amazon DocumentDB possède un point de terminaison d'instance par instance active.

Un point de terminaison d'instance exerce un contrôle direct sur une instance spécifique, pour les scénarios où l'utilisation du point de terminaison de cluster ou du point de terminaison de lecteur peut ne pas être appropriée. Un exemple de cas d'utilisation est la mise en service pour une charge de travail périodique des analyses en lecture seule. Vous pouvez provisionner une instance de larger-than-normal réplique, vous connecter directement à la nouvelle instance plus grande avec son point de terminaison, exécuter les requêtes d'analyse, puis mettre fin à l'instance. L'utilisation du point de terminaison d'instance empêche le trafic des analyses d'avoir un effet sur d'autres instances de cluster.

Voici un exemple de point de terminaison d'instance pour une instance unique dans un cluster Amazon DocumentDB :

```
sample-instance.123456789012.us-east-1.docdb.amazonaws.com:27017
```

L'exemple suivant présente une chaîne de connexion utilisant ce point de terminaison d'instance :

```
mongodb://username:password@sample-instance.123456789012.us-
east-1.docdb.amazonaws.com:27017
```

> **Note**
>
> Un rôle de l'instance en tant que principale ou réplica peut changer suite à un événement de basculement. Vos applications ne doivent jamais supposer qu'un point de terminaison d'une instance particulière est le principal. Nous ne recommandons pas la connexion aux points de terminaison d'instance pour les applications en production. Au lieu de cela, nous vous recommandons de vous connecter à votre cluster à l'aide du point de terminaison du cluster et en mode jeu de réplicas (voir Connexion à Amazon DocumentDB en tant qu'ensemble

de réplicas). Pour un contrôle plus avancé de la priorité de basculement d'une instance, consultez Comprendre la tolérance aux pannes des clusters Amazon DocumentDB.

Pour plus d'informations sur la recherche des points de terminaison d'un cluster, consultez Recherche d'un point de terminaison de l'instance.

Mode Jeux de réplicas

Vous pouvez vous connecter à votre point de terminaison de cluster Amazon DocumentDB en mode jeu de répliques en spécifiant le nom du jeu de répliques. `rs0` La connexion en mode Jeu de réplicas fournit la capacité de spécifier les options Problème de lecture, Problème d'écriture et Préférences de lecture. Pour plus d'informations, consultez Cohérence en lecture.

Voici un exemple de chaîne de connexion se connectant en mode Jeu de réplicas :

```
mongodb://username:password@sample-cluster.cluster-123456789012.us-
east-1.docdb.amazonaws.com:27017/?replicaSet=rs0
```

Lorsque vous vous connectez en mode jeu de répliques, votre cluster Amazon DocumentDB apparaît à vos pilotes et clients sous la forme d'un jeu de répliques. Les instances ajoutées et supprimées de votre cluster Amazon DocumentDB sont automatiquement reflétées dans la configuration du jeu de répliques.

Chaque cluster Amazon DocumentDB se compose d'un seul ensemble de répliques portant le nom par défaut. `rs0` Le nom du jeu de réplicas ne peut pas être modifié.

La connexion au point de terminaison du cluster en mode Jeu de réplicas est la méthode recommandée pour une utilisation générale.

> **Note**
>
> Toutes les instances d'un cluster Amazon DocumentDB écoutent les connexions sur le même port TCP.

TLS Support

Pour plus d'informations sur la connexion à Amazon DocumentDB à l'aide du protocole TLS (Transport Layer Security), consultez. Chiffrement des données en transit

Stockage Amazon DocumentDB

Les données Amazon DocumentDB sont stockées dans un volume de cluster, qui est un volume virtuel unique qui utilise des disques SSD. Un volume de cluster se compose de six copies de vos données, qui sont répliquées automatiquement sur plusieurs zones de disponibilité en une seule Région AWS. Cette réplication garantit que vos données sont hautement durables, avec une possibilité moindre de perte des données. Elle permet également de vous assurer que votre cluster est plus disponible pendant un basculement, car les copies de vos données existent déjà dans d'autres zones de disponibilité. Ces copies peuvent continuer à envoyer des demandes de données aux instances de votre cluster Amazon DocumentDB.

Facturation du stockage des données

Amazon DocumentDB augmente automatiquement la taille d'un volume de cluster à mesure que la quantité de données augmente. Un volume de cluster Amazon DocumentDB peut atteindre une taille maximale de 128 TiB ; toutefois, seul l'espace que vous utilisez dans un volume de cluster Amazon DocumentDB vous est facturé. À partir d'Amazon DocumentDB 4.0, lorsque des données sont supprimées, par exemple en supprimant une collection ou un index, l'espace global alloué diminue d'une quantité comparable. Ainsi, vous pouvez réduire les frais de stockage en supprimant les collections, les index et les bases de données dont vous n'avez plus besoin. Avec Amazon DocumentDB 3.6, lorsque des données sont supprimées, par exemple en supprimant une collection ou un index, l'espace global alloué reste le même. L'espace libre est réutilisé automatiquement lorsque le volume de données augmente à l'avenir.

> **Note**
>
> Avec Amazon DocumentDB 3.6, les coûts de stockage sont basés sur le « seuil maximum » de stockage (le montant maximum alloué au cluster Amazon DocumentDB à un moment donné). Vous pouvez gérer les coûts en évitant les pratiques ETL qui créent de gros volumes d'informations temporaires ou qui chargent de gros volumes de nouvelles données avant de supprimer les anciennes données inutiles. Si la suppression de données d'un cluster Amazon DocumentDB se traduit par une quantité importante d'espace alloué mais inutilisé, la réinitialisation du seuil maximum nécessite d'effectuer un vidage logique des données et de les restaurer sur un nouveau cluster, à l'aide d'un outil tel que ou. `mongodump` `mongorestore` Le fait de créer et de restaurer un instantané n'a pas pour effet de réduire le stockage alloué, car la structure physique du stockage sous-jacent reste inchangée dans l'instantané restauré.

> **Note**
>
> Le recours aux utilitaires comme `mongodump` et `mongorestore` entraîne des frais d'E/S en fonction de la taille des données lues et écrites sur le volume de stockage.

Pour plus d'informations sur le stockage des données et la tarification des E/S d'Amazon DocumentDB, consultez les FAQ sur les tarifs et les tarifs d'Amazon DocumentDB (compatible avec MongoDB).

Réplication Amazon DocumentDB

Dans un cluster Amazon DocumentDB, chaque instance de réplique expose un point de terminaison indépendant. Ces points de terminaison de réplica fournissent l'accès en lecture seule aux données du volume de cluster. Ils vous permettent de dimensionner la charge de travail en lecture pour vos données sur plusieurs instances répliquées. Ils contribuent également à améliorer les performances de lecture des données et à augmenter la disponibilité des données dans votre cluster Amazon DocumentDB. Les répliques Amazon DocumentDB sont également des cibles de basculement et sont rapidement promues en cas de défaillance de l'instance principale de votre cluster Amazon DocumentDB.

Fiabilité d'Amazon DocumentDB

Amazon DocumentDB est conçu pour être fiable, durable et tolérant aux pannes. (Pour améliorer la disponibilité, vous devez configurer votre cluster Amazon DocumentDB de manière à ce qu'il dispose de plusieurs instances de réplication dans différentes zones de disponibilité.) Amazon DocumentDB inclut plusieurs fonctionnalités automatiques qui en font une solution de base de données fiable.

Réparation automatique du stockage

Amazon DocumentDB conserve plusieurs copies de vos données dans trois zones de disponibilité, ce qui réduit considérablement le risque de perte de données en cas de panne de stockage. Amazon DocumentDB détecte automatiquement les défaillances dans le volume du cluster. Lorsqu'un segment d'un volume de cluster tombe en panne, Amazon DocumentDB répare immédiatement le segment. Il utilise les données des autres volumes qui composent le volume de cluster pour garantir que les données du segment réparé sont actives. Amazon DocumentDB évite ainsi les pertes de données et réduit le besoin d'effectuer une point-in-time restauration pour récupérer après une défaillance d'instance.

Préparation du cache « survivable »

Amazon DocumentDB gère son cache de pages dans le cadre d'un processus distinct de celui de la base de données afin que le cache de pages puisse survivre indépendamment de la base de données. Dans l'éventualité peu probable d'une défaillance de la base de données, le cache de page reste en mémoire. Cela garantit que le groupe de tampons est préparé avec l'état le plus courant au redémarrage de la base de données.

Récupération sur incident

Amazon DocumentDB est conçu pour effectuer une restauration quasi instantanée en cas de panne et pour continuer à diffuser les données de votre application. Amazon DocumentDB effectue une restauration après incident de manière asynchrone sur des threads parallèles afin que votre base de données soit ouverte et disponible presque immédiatement après un crash.

Gouvernance des ressources

Amazon DocumentDB protège les ressources nécessaires à l'exécution des processus critiques du service, tels que les bilans de santé. Pour ce faire, et lorsqu'une instance est confrontée à une pression de mémoire élevée, Amazon DocumentDB limite les demandes. Par conséquent, certaines opérations peuvent être mises en file d'attente pour attendre que la pression sur la mémoire diminue. Si la pression sur la mémoire persiste, les opérations en file d'attente peuvent expirer. Vous pouvez vérifier si le service ralentit les opérations en raison d'un manque de mémoire à l'aide des CloudWatch mesures suivantes :`LowMemThrottleQueueDepth`,`LowMemThrottleMaxQueueDepth`,`LowMemNumOperationsT` `LowMemNumOperationsTimedOut` Pour plus d'informations, consultez la section Surveillance d'Amazon DocumentDB avec. CloudWatch Si vous constatez une pression de mémoire soutenue sur votre instance en raison de ces LowMem CloudWatch indicateurs, nous vous conseillons de la dimensionner afin de fournir de la mémoire supplémentaire pour votre charge de travail.

Options de préférence de lecture

Amazon DocumentDB utilise un service de stockage partagé cloud natif qui réplique les données six fois sur trois zones de disponibilité afin de garantir des niveaux de durabilité élevés. Amazon DocumentDB ne repose pas sur la réplication de données vers plusieurs instances pour garantir la durabilité. Les données de votre cluster sont durable qu'elles contiennent une seule instance ou 15 instances.

Durabilité en écriture

Amazon DocumentDB utilise un système de stockage unique, distribué, tolérant aux pannes et autoréparateur. Ce système réplique six copies (V=6) de vos données dans trois zones de disponibilité pour garantir une AWS disponibilité et une durabilité élevées. Lors de l'écriture de données, Amazon DocumentDB s'assure que toutes les écritures sont enregistrées de manière durable sur la majorité des nœuds avant de confirmer l'écriture au client. Si vous utilisez un jeu de répliques MongoDB à trois nœuds, l'utilisation d'un souci d'écriture `{w:3, j:true}` de permettrait d'obtenir la meilleure configuration possible par rapport à Amazon DocumentDB.

Les écritures vers un cluster Amazon DocumentDB doivent être traitées par l'instance d'écriture du cluster. Toute tentative d'écriture dans un lecteur entraîne une erreur. Une écriture confirmée depuis une instance principale Amazon DocumentDB est durable et ne peut pas être annulée. Amazon DocumentDB est très durable par défaut et ne prend pas en charge les options d'écriture non durables. Vous ne pouvez pas modifier le niveau de durabilité (c'est-à-dire, le problème d'écriture). Amazon DocumentDB ignore w=anything et affiche effectivement w : 3 et j : true. Vous ne pouvez pas le réduire.

Le stockage et le calcul étant séparés dans l'architecture Amazon DocumentDB, un cluster avec une seule instance est extrêmement durable. La durabilité est gérée au niveau de la couche de stockage. Par conséquent, un cluster Amazon DocumentDB avec une seule instance et un cluster avec trois instances atteint le même niveau de durabilité. Vous pouvez configurer votre cluster pour votre cas d'utilisation spécifique tout en fournissant une durabilité élevée pour vos données.

Les écritures vers un cluster Amazon DocumentDB sont atomiques au sein d'un même document.

Amazon DocumentDB ne prend pas en charge `wtimeout` cette option et ne renverra pas d'erreur si une valeur est spécifiée. Il est garanti que les écritures sur l'instance Amazon DocumentDB principale ne seront pas bloquées indéfiniment.

Isolation en écriture

Les lectures effectuées à partir d'une instance Amazon DocumentDB ne renvoient que des données durables avant le début de la requête. Les lectures ne renvoient jamais des données modifiées après la début de l'exécution par la requête et des lectures sales ne sont pas possibles, quelles que soient les circonstances.

Cohérence en lecture

Les données lues depuis un cluster Amazon DocumentDB sont durables et ne seront pas annulées. Vous pouvez modifier la cohérence de lecture pour les lectures Amazon DocumentDB en spécifiant la préférence de lecture pour la demande ou la connexion. Amazon DocumentDB ne prend pas en charge les options de lecture non durables.

Les lectures effectuées à partir de l'instance principale d'un cluster Amazon DocumentDB sont parfaitement cohérentes dans des conditions de fonctionnement normales et sont read-after-write cohérentes. Si un basculement se produit entre la lecture et l'écriture ultérieure, le système peut brièvement renvoyer une lecture qui n'est pas fortement cohérente. Toutes les lectures à partir d'un réplica en lecture présentent une cohérence éventuelle et renvoient les données dans le même ordre, et souvent avec une latence de réplica inférieure à 100 millisecondes.

Préférences de lecture d'Amazon DocumentDB

Amazon DocumentDB prend en charge la définition d'une option de préférence de lecture uniquement lors de la lecture de données depuis le point de terminaison du cluster en mode Replica Set. La définition d'une option de préférence de lecture affecte la manière dont votre client ou pilote MongoDB achemine les demandes de lecture vers les instances de votre cluster Amazon DocumentDB. Vous pouvez définir des options de préférence de lecture pour une requête spécifique, ou en tant qu'option générale dans votre pilote MongoDB. (Consultez votre client ou la documentation du pilote pour obtenir des instructions sur la façon de définir une option de préférence de lecture).

Si votre client ou pilote ne se connecte pas à un point de terminaison du cluster Amazon DocumentDB en mode Replica Set, le résultat de la spécification d'une préférence de lecture n'est pas défini.

Amazon DocumentDB ne prend pas en charge la définition de jeux de balises comme préférence de lecture.

Options de préférences de lecture prises en charge

- **`primary`**—La spécification d'une préférence de `primary` lecture permet de garantir que toutes les lectures sont acheminées vers l'instance principale du cluster. Si l'instance principale n'est pas disponible, l'opération de lecture échoue. Une préférence de `primary` lecture garantit la read-after-write cohérence et convient aux cas d'utilisation qui privilégient la read-after-write cohérence par rapport à la haute disponibilité et à la mise à l'échelle de lecture.

  L'exemple suivant spécifie une préférence de lecture « `primary` » :

```
db.example.find().readPref('primary')
```

- **`primaryPreferred`**—La spécification d'une préférence de `primaryPreferred` lecture achemine les lectures vers l'instance principale dans le cadre d'un fonctionnement normal. En cas de basculement principal, le client achemine les demandes vers un réplica. Une préférence de `primaryPreferred` lecture garantit read-after-write la cohérence pendant le fonctionnement normal et, en fin de compte, la cohérence des lectures lors d'un événement de basculement. Une préférence de `primaryPreferred` lecture convient aux cas d'utilisation qui privilégient la read-after-write cohérence par rapport à la mise à l'échelle de lecture, mais qui nécessitent tout de même une haute disponibilité.

  L'exemple suivant spécifie une préférence de lecture « `primaryPreferred` » :

  ```
  db.example.find().readPref('primaryPreferred')
  ```

- **`secondary`**—La spécification d'une préférence de `secondary` lecture garantit que les lectures ne sont acheminées que vers une réplique, jamais vers l'instance principale. S'il n'y a pas d'instances de réplica dans un cluster, la demande de lecture échoue. Une préférence de `secondary` lecture aboutit à des lectures cohérentes et convient aux cas d'utilisation qui privilégient le débit d'écriture de l'instance principale par rapport à la haute disponibilité et à la read-after-write cohérence.

  L'exemple suivant spécifie une préférence de lecture « `secondary` » :

  ```
  db.example.find().readPref('secondary')
  ```

- **`secondaryPreferred`**—La spécification d'une préférence de `secondaryPreferred` lecture garantit que les lectures sont acheminées vers une réplique en lecture lorsqu'une ou plusieurs répliques sont actives. S'il n'y a pas d'instances de réplica actives dans un cluster, la demande de lecture est acheminé vers l'instance principale. Une préférence de lecture « `secondaryPreferred` » génère des lectures cohérentes à terme (eventually consistent) lorsque la lecture est traitée par un réplica en lecture. Cela permet d'obtenir de la read-after-write cohérence lorsque la lecture est prise en charge par l'instance principale (sauf en cas de basculement). Une préférence de `secondaryPreferred` lecture convient aux cas d'utilisation qui

privilégient le dimensionnement de la lecture et la haute disponibilité plutôt que la read-after-write cohérence.

L'exemple suivant spécifie une préférence de lecture « `secondaryPreferred` » :

```
db.example.find().readPref('secondaryPreferred')
```

- **`nearest`**—La spécification d'une préférence de `nearest` lecture achemine les lectures uniquement en fonction de la latence mesurée entre le client et toutes les instances du cluster Amazon DocumentDB. Une préférence de lecture « `nearest` » génère des lectures cohérentes à terme (eventually consistent) lorsque la lecture est traitée par un réplica en lecture. Cela permet d'obtenir de la read-after-write cohérence lorsque la lecture est prise en charge par l'instance principale (sauf en cas de basculement). Une préférence de `nearest` lecture convient aux cas d'utilisation qui privilégient la latence de lecture la plus faible possible et la haute disponibilité plutôt que la read-after-write cohérence et la mise à l'échelle de lecture.

  L'exemple suivant spécifie une préférence de lecture « `nearest` » :

```
db.example.find().readPref('nearest')
```

Haute disponibilité

Amazon DocumentDB prend en charge les configurations de clusters à haute disponibilité en utilisant des répliques comme cibles de basculement pour l'instance principale. En cas de défaillance de l'instance principale, une réplique Amazon DocumentDB est promue en tant que nouvelle instance principale, avec une brève interruption au cours de laquelle les demandes de lecture et d'écriture adressées à l'instance principale échouent, sauf exception.

Si votre cluster Amazon DocumentDB n'inclut aucune réplique, l'instance principale est recréée en cas de panne. Cependant, la promotion d'une réplique Amazon DocumentDB est beaucoup plus rapide que la recréation de l'instance principale. Nous vous recommandons donc de créer une ou plusieurs répliques Amazon DocumentDB comme cibles de basculement.

Les réplicas qui sont destinés à être utilisés comme cibles de basculement doivent appartenir à la même classe d'instance que l'instance principale. Ils doivent être mis en service dans des zones de disponibilité autres que celle de l'instance principale. Vous pouvez décider les réplicas préférés comme cibles de basculement. Pour connaître les meilleures pratiques relatives à la configuration

d'Amazon DocumentDB pour une haute disponibilité, consultez. Comprendre la tolérance aux pannes des clusters Amazon DocumentDB

Lectures de dimensionnement

Les répliques Amazon DocumentDB sont idéales pour le dimensionnement des lectures. Elles sont entièrement dédiées aux opérations de lecture sur votre volume de cluster, ce qui signifie que les réplicas ne traitent pas les écritures. La réplication de données se produit au sein du volume de cluster et non pas entre les instances. Par conséquent, les ressources de chaque réplica sont dédiées au traitement de vos requêtes, et non à la réplication et à l'écriture des données.

Si votre application exige plus de capacité de lecture, vous pouvez rapidement ajouter un réplica à votre cluster (généralement en moins de 10 minutes). Si vos exigences en matière de capacités en lecture diminuent, vous pouvez supprimer les réplicas devenus inutiles. Avec les répliques Amazon DocumentDB, vous ne payez que pour la capacité de lecture dont vous avez besoin.

Amazon DocumentDB prend en charge le dimensionnement de lecture côté client grâce à l'utilisation des options de préférence de lecture. Pour plus d’informations, consultez Préférences de lecture d'Amazon DocumentDB.

Suppressions TTL

Les suppressions depuis une zone d'index TTL via un processus en arrière-plan sont effectuées dans la mesure du possible et ne sont pas garanties au cours d'une période spécifique. Des facteurs tels que la taille des instances, l'utilisation de ressources des instances, la taille de document et le débit global peuvent affecter le déroulement d'une suppression TTL.

Lorsque le moniteur TTL supprime vos documents, chaque suppression entraîne des coûts d'E/S, ce qui augmente le montant de votre facture. Si les frais de suppression de débit et de TTL augmentent, vous devez vous attendre à une augmentation de votre facture, étant donné que l'utilisation des E/S augmente.

Lorsque vous créez un index TTL sur une collection existante, vous devez supprimer tous les documents expirés avant de créer l'index. L'implémentation TTL actuelle est optimisée pour supprimer une petite partie des documents de la collection, ce qui est typique si le TTL a été activé sur la collection dès le début, et peut entraîner des IOPS plus élevées que nécessaire si un grand nombre de documents doivent être supprimés en une seule fois.

Si vous ne souhaitez pas créer d'index TTL pour supprimer des documents, vous pouvez segmenter les documents en collections en fonction du temps et simplement supprimer ces collections lorsque

les documents ne sont plus nécessaires. Par exemple : vous pouvez créer une collection par semaine et la supprimer sans encourir de frais d'E/S. Cela peut être nettement plus rentable que l'utilisation d'un indice TTL.

Ressources facturables

Identification des ressources Amazon DocumentDB facturables

En tant que service de base de données entièrement géré, Amazon DocumentDB facture les instances, le stockage, les E/S, les sauvegardes et le transfert de données. Pour plus d'informations, consultez la tarification d'Amazon DocumentDB (compatible avec MongoDB).

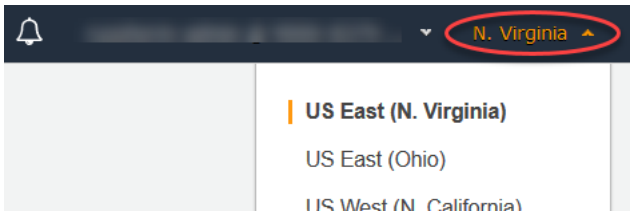
Pour découvrir les ressources facturables présentes sur votre compte et éventuellement les supprimer, vous pouvez utiliser le AWS Management Console ou AWS CLI.

À l'aide du AWS Management Console

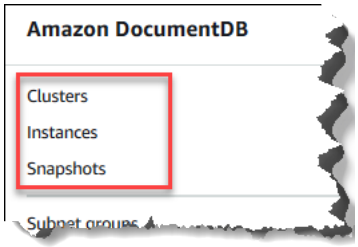
À l'aide de AWS Management Console, vous pouvez découvrir les clusters, les instances et les instantanés Amazon DocumentDB que vous avez provisionnés pour un usage donné. Région AWS

Pour découvrir des clusters, instances et instantanés

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Pour découvrir les ressources facturables dans une région autre que votre région par défaut, dans le coin supérieur droit de l'écran, choisissez Région AWS celle que vous souhaitez rechercher.



3. Dans le panneau de navigation, choisissez le type de ressource facturable qui vous intéresse : Clusters, Instances ou Snapshots (Instantanés).



4. Tous vos clusters,instances ou instantanés mis en service pour la région sont répertoriés dans le panneau droit. Vous serez facturé pour les clusters, les instances et les instantanés.

À l'aide du AWS CLI

À l'aide de AWS CLI, vous pouvez découvrir les clusters, les instances et les instantanés Amazon DocumentDB que vous avez provisionnés pour un usage donné. Région AWS

Pour découvrir des clusters et des instances

Le code suivant répertorie tous vos clusters et instances pour la région spécifiée. Si vous souhaitez rechercher des clusters et des instances dans votre région par défaut, vous pouvez omettre le paramètre `--region`.

Example

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-clusters \
    --region us-east-1 \
    --query 'DBClusters[?Engine==`docdb`]' | \
       grep -e "DBClusterIdentifier" -e "DBInstanceIdentifier"
```

Pour Windows :

```
aws docdb describe-db-clusters ^
    --region us-east-1 ^
    --query 'DBClusters[?Engine==`docdb`]' | ^
       grep -e "DBClusterIdentifier" -e "DBInstanceIdentifier"
```

Le résultat de cette opération ressemble à ceci.

```
"DBClusterIdentifier": "docdb-2019-01-09-23-55-38",
        "DBInstanceIdentifier": "docdb-2019-01-09-23-55-38",
        "DBInstanceIdentifier": "docdb-2019-01-09-23-55-382",
"DBClusterIdentifier": "sample-cluster",
"DBClusterIdentifier": "sample-cluster2",
```

Pour découvrir des instantanés

Le code suivant répertorie tous vos instantanés pour la région spécifiée. Si vous souhaitez rechercher des instantanés dans votre région par défaut, vous pouvez omettre le paramètre `--region`.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-cluster-snapshots \
  --region us-east-1 \
  --query 'DBClusterSnapshots[?Engine==`docdb`].
[DBClusterSnapshotIdentifier,SnapshotType]'
```

Pour Windows :

```
aws docdb describe-db-cluster-snapshots ^
  --region us-east-1 ^
  --query 'DBClusterSnapshots[?Engine==`docdb`].
[DBClusterSnapshotIdentifier,SnapshotType]'
```

Le résultat de cette opération ressemble à ceci.

```
[
    [
        "rds:docdb-2019-01-09-23-55-38-2019-02-13-00-06",
        "automated"
    ],
    [
        "test-snap",
        "manual"
    ]
]
```

Il vous suffit de supprimer les instantanés `manual`. Les instantanés`Automated` sont supprimés lorsque vous supprimez le cluster.

Suppression de ressources facturables indésirables

Pour supprimer un cluster, vous devez d'abord supprimer toutes les instances du cluster.

- Pour supprimer des instances, consultez Supprimer une instance Amazon DocumentDB .

> **Important**
>
> Même si vous supprimez les instances d'un cluster, vous êtes encore facturé pour l'utilisation du stockage et des sauvegardes associée à ce cluster. Pour arrêter tous les frais, vous devez également supprimer votre cluster et les instantanés manuels.

- Pour supprimer des clusters, consultez Suppression d'un cluster Amazon DocumentDB.
- Pour supprimer des instantanés manuels, veuillez consulter Suppression d'un instantané de cluster.

Qu'est-ce qu'une base de données de documents ?

Certains développeurs ne considèrent pas leur modèle de données en termes de lignes et de colonnes normalisées. En général, dans le niveau de l'application, les données sont représentées comme un document JSON parce qu'il est plus intuitif pour les développeurs de considérer leur modèle de données comme un document.

La popularité des bases de données de documents s'est accrue parce qu'elles permettent de conserver des données dans une base de données en utilisant le même format de modèle de document qu'on utilise dans le code d'application. Les bases de données de documents fournissent des API puissantes et intuitives pour un développement flexible et agile.

Rubriques

- Cas d'utilisation de base de données de documents
- Compréhension des documents
- Travailler avec des documents

Cas d'utilisation de base de données de documents

Selon votre cas d'utilisation, vous aurez besoin d'une base de données de documents ou d'un autre type de base de données pour la gestion de vos données. Les bases de données de documents sont utiles pour les charges de travail qui nécessitent un schéma flexible et rapide, et un développement itératif. Voici quelques exemples de cas d'utilisation pour lesquels les bases de données de documents peuvent offrir des avantages significatifs :

Rubriques

- Profils d'utilisateurs
- Big Data en temps réel
- Gestion de contenu

Profils d'utilisateurs

Les bases de données de documents présentent un schéma flexible, elles peuvent donc stocker des documents qui ont différents attributs et valeurs de données. Les bases de données de documents constituent une solution pratique pour les profils en ligne dans lesquels différents utilisateurs fournissent différents types d'informations. L'utilisation d'une base de données de documents, vous permet de stocker efficacement le profil de chaque utilisateur en stockant uniquement les attributs qui sont spécifiques à chacun d'eux.

Supposons qu'un utilisateur choisisse d'ajouter ou de supprimer des informations dans son profil. Dans ce cas, son document peut être facilement remplacé par une version mise à jour qui contient des attributs et des données récemment ajoutés ou nouvellement omises. Les bases de données de documents gèrent facilement ce niveau d'individualité et de fluidité.

Big Data en temps réel

Historiquement, la possibilité d'extraire des informations à partir de données opérationnelles a été entravée par le fait que les bases de données opérationnelles et bases de données analytiques étaient maintenues dans des environnements différents opérationnels et professionnels/reporting respectivement. Dans un environnement professionnel très compétitif, la capacité d'extraire des informations opérationnelles en temps réel est critique. En utilisant des bases de données de documents, une entreprise peut stocker et gérer des données opérationnelles à partir de n'importe quelle source et envoyer simultanément les données vers le moteur de BI choisi pour l'analyse. Il n'est pas nécessaire d'avoir deux environnements.

Gestion de contenu

Pour gérer de manière efficace le contenu, vous devez être en mesure de collecter et regrouper du contenu à partir d'une variété de sources, pour ensuite le délivrer au client. Grâce à leur schéma flexible, les bases de données de documents sont parfaites pour collecter et stocker tous les types de données. Vous pouvez les utiliser pour créer et intégrer de nouveaux types de contenu, y compris le contenu généré par l'utilisateur, notamment des images, des commentaires et des vidéos.

Compréhension des documents

Les bases de données de documents sont utilisées pour stocker des données semi-structurées sous la forme d'un document au lieu de normaliser les données sur plusieurs tables, chacune avec une structure propre et fixe, comme dans une base de données relationnelle. Les documents stockés dans une base de données de documents utilisent des paires clé-valeur imbriquées pour fournir la structure ou le schéma du document. Cependant, différents types de documents peuvent être stockés dans une même base de données de documents, pour satisfaire ainsi à l'exigence de traitement des données similaires dans des formats différents. Par exemple, chaque document étant auto-descriptif, les documents codés au format JSON pour une boutique en ligne qui sont décrits dans la rubrique Exemple de documents dans une base de données de documents peuvent être stockés dans la même base de données de documents.

Rubriques

Terminologie non relationnelle et SQL et

Le tableau suivant compare la terminologie utilisée par certaines bases de données de documents (MongoDB) avec la terminologie utilisée par les bases de données SQL.

SQL	MongoDB
Tableau	Collection
Rangée	Document
Colonne	Champ
Clé primaire	ObjectId
Index	Index

SQL	MongoDB
Vue	Vue
Table ou objet imbriqué	Document intégré
Tableau	Tableau

Documents simples

Tous les documents d'une base de données de documents sont auto-descriptifs. Cette documentation utilise des documents au format JSON, mais vous pouvez utiliser d'autres moyens de codage.

Un document simple possède un ou plusieurs champs qui se trouvent tous au même niveau dans le document. Dans l'exemple suivant, les champs `SSN`, `LName`, `FName`, `DOB`, `Street`, `City`, `State-Province`, `PostalCode`, et `Country` sont tous des parents dans le document.

```
{
   "SSN": "123-45-6789",
   "LName": "Rivera",
   "FName": "Martha",
   "DOB": "1992-11-16",
   "Street": "125 Main St.",
   "City": "Anytown",
   "State-Province": "WA",
   "PostalCode": "98117",
   "Country": "USA"
}
```

Lorsque les informations sont organisées dans un document simple, chaque champ est gérée de manière individuelle. Pour récupérer l'adresse d'une personne, vous devez extraire `Street`, `City`, `State-Province`, `PostalCode`, et `Country` comme des éléments de données individuels.

Documents intégrés

Un document complexe organise ses données en créant des documents intégrés dans le document. Les documents intégrés permettent de gérer les données au sein de regroupements et en tant qu'éléments de données individuels, selon ce qui est le plus efficace dans un cas donné. En utilisant

l'exemple précédent, vous pouvez intégrer un document `Address` dans le document principal. Cette action se traduit par la structure de document suivante :

```
{
   "SSN": "123-45-6789",
   "LName": "Rivera",
   "FName": "Martha",
   "DOB": "1992-11-16",
   "Address":
   {
       "Street": "125 Main St.",
       "City": "Anytown",
       "State-Province": "WA",
       "PostalCode": "98117",
       "Country": "USA"
   }
}
```

Vous pouvez désormais accéder aux données dans le document sous la forme de champs individuels (`"SSN":`), en tant que document intégré (`"Address":`), ou en tant que membre d'un document incorporé (`"Address":{"Street":}`).

Exemple de documents dans une base de données de documents

Comme indiqué précédemment, chaque document d'une base de données documents est auto-descriptif, par conséquent la structure des documents au sein d'une base de données de documents peut être différente de l'un à l'autre. Les deux documents suivants, l'un pour un livre et l'autre pour une revue, ont des structures différentes. Cependant, tous les deux peuvent se trouver dans la même base de données de documents.

Voici un exemple de document de livre :

```
{
    "_id" : "9876543210123",
    "Type": "book",
    "ISBN": "987-6-543-21012-3",
    "Author":
    {
        "LName":"Roe",
        "MI": "T",
        "FName": "Richard"
```

```
    },
    "Title": "Understanding Document Databases"
}
```

Voici un exemple de document de revue avec deux articles :

```
{
    "_id" : "0123456789012",
    "Publication": "Programming Today",
    "Issue":
    {
        "Volume": "14",
        "Number": "09"
    },
    "Articles" : [
        {
            "Title": "Is a Document Database Your Best Solution?",
            "Author":
            {
                "LName": "Major",
                "FName": "Mary"
            }
        },
        {
            "Title": "Databases for Online Solutions",
            "Author":
            {
                "LName": "Stiles",
                "FName": "John"
            }
        }
    ],
    "Type": "periodical"
}
```

Comparez la structure de ces deux documents. Avec une base de données relationnelle, vous devez soit séparer les tables « périodique » et « livres », soit avoir une seule table avec des champs non utilisés, par exemple « Publication », « Numéro », « Articles » et « MI », en tant que valeurs `null`. Les bases de données de documents sont semi-structurées, chaque document définit sa propre structure, par conséquent ces deux documents peuvent coexister dans la même base de données de documents sans champs `null`. Les bases de données de documents facilitent le traitement de données fragmentées.

Le développement à partir d'une base de données de documents permet un développement rapide et itératif. En effet, vous pouvez modifier la structure des données d'un document de manière dynamique, sans devoir modifier le schéma pour la totalité de la collection. Les bases de données de documents sont idéales pour le développement souple et les environnements dynamiques et évolutifs.

Compréhension de la normalisation dans une base de données de documents

Les bases de données de documents ne sont pas normalisées. Les données trouvées dans un document peut être répétées dans un autre document. De plus, il peut exister certaines divergences de données entre les documents. Par exemple, prenez le scénario dans lequel vous effectuez un achat dans une boutique en ligne et tous les détails de vos achats sont stockés dans un seul document. Le document peut ressembler au document JSON suivant :

```
{
    "DateTime": "2018-08-15T12:13:10Z",
    "LName" : "Santos",
    "FName" : "Paul",
    "Cart" : [
        {
            "ItemId" : "9876543210123",
            "Description" : "Understanding Document Databases",
            "Price" : "29.95"
        },
        {
            "ItemId" : "0123456789012",
            "Description" : "Programming Today",
            "Issue": {
                "Volume": "14",
                "Number": "09"
            },
            "Price" : "8.95"
        },
        {
            "ItemId": "234567890-K",
            "Description": "Gel Pen (black)",
            "Price": "2.49"
        }
    ],
    "PaymentMethod" :
    {
        "Issuer" : "MasterCard",
```

```
        "Number" : "1234-5678-9012-3456"
    },
    "ShopperId" : "1234567890"
}
```

Toutes ces informations sont stockées en tant que document dans une collection de transactions. Par la suite, vous vous rendez compte que vous avez oublié d'acheter un élément. Par conséquent, vous vous connectez à la même boutique et vous effectuez un autre achat, lequel est également stocké en tant qu'un autre document dans la collection de transactions.

```
{
    "DateTime": "2018-08-15T14:49:00Z",
    "LName" : "Santos",
    "FName" : "Paul",
    "Cart" : [
        {
            "ItemId" : "2109876543210",
            "Description" : "Document Databases for Fun and Profit",
            "Price" : "45.95"
        }
    ],
    "PaymentMethod" :
    {
        "Issuer" : "Visa",
        "Number" : "0987-6543-2109-8765"
    },
    "ShopperId" : "1234567890"
}
```

Il convient de noter la redondance entre ces deux documents : votre nom et votre ID d'acheteur (et, si vous avez utilisé la même carte de paiement, les informations relative à votre carte de paiement). C'est acceptable, car le stockage est peu coûteux, et chaque document enregistre complètement une seule transaction que l'on peut récupérer rapidement avec une simple requête clé-valeur qui ne nécessite pas de jonctions.

Il existe également une différence apparente entre les deux documents : les informations relatives à votre carte de paiement. Il s'agit uniquement d'une différence apparente, car vous avez probablement utilisé une autre carte de paiement différente pour chaque achat. Chaque document est approprié pour la transaction qu'il documente.

Travailler avec des documents

En tant que base de données de documents, Amazon DocumentDB facilite le stockage, l'interrogation et l'indexation de données JSON. Dans Amazon DocumentDB, une collection est analogue à une table dans une base de données relationnelle, sauf qu'il n'existe pas de schéma unique appliqué à tous les documents. Les collections vous permettent de regrouper des documents similaires, tout en les conservant tous dans la même base de données, sans que leur structure doive être identique.

En utilisant les documents d'exemple des rubriques précédentes, il est probable que vous ayez des collections pour `reading_material` et `office_supplies`. Votre logiciel est responsable d'associer un document avec la collection à laquelle il appartient.

Les exemples suivants utilisent l'API MongoDB pour montrer comment ajouter, interroger, mettre à jour et supprimer des documents.

Rubriques

Ajouter des documents

Dans Amazon DocumentDB, une base de données est créée lorsque vous ajoutez pour la première fois un document à une collection. Dans cet exemple, vous créez une collection nommée `example` dans la base de données `test`, qui est la base de données par défaut lorsque vous vous connectez à un cluster. Étant donné que la collection est implicitement créée lorsque le premier document est inséré, il n'y a pas de vérification d'erreur relative au nom de la collection. Par conséquent, une faute de frappe dans le nom de la collection, telle que `eexample` au lieu de `example`, créera et ajoutera le document à la collection `eexample` au lieu de l'ajouter à la collection souhaitée. Votre application est chargée de la vérification des erreurs.

Les exemples suivants utilisent l'API MongoDB pour ajouter des documents.

Rubriques

- Ajouter plusieurs documents

Ajouter un seul document

Pour ajouter un seul document à une collection, utilisez l'opération `insertOne( {} )` avec le document que vous souhaitez ajouter à la collection.

```
db.example.insertOne(
    {
        "Item": "Ruler",
        "Colors": ["Red","Green","Blue","Clear","Yellow"],
        "Inventory": {
            "OnHand": 47,
            "MinOnHand": 40
        },
        "UnitPrice": 0.89
    }
)
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "acknowledged" : true,
    "insertedId" : ObjectId("5bedafbcf65ff161707de24f")
}
```

Ajouter plusieurs documents

Pour ajouter plusieurs documents à une collection, utilisez l'opération `insertMany( [{},..., {}] )` avec une liste des document que vous souhaitez ajouter à la collection. Bien que dans cette liste particulière, les documents ont des schémas différents, ils peuvent tous être ajoutés à la même collection.

```
db.example.insertMany(
    [
        {
            "Item": "Pen",
            "Colors": ["Red","Green","Blue","Black"],
            "Inventory": {
                "OnHand": 244,
                "MinOnHand": 72
```

```
            }
        },
        {
            "Item": "Poster Paint",
            "Colors": ["Red","Green","Blue","Black","White"],
            "Inventory": {
                "OnHand": 47,
                "MinOnHand": 50
            }
        },
        {
            "Item": "Spray Paint",
            "Colors": ["Black","Red","Green","Blue"],
            "Inventory": {
                "OnHand": 47,
                "MinOnHand": 50,
                "OrderQnty": 36
            }
        }
    ]
)
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "acknowledged" : true,
    "insertedIds" : [
            ObjectId("5bedb07941ca8d9198f5934c"),
            ObjectId("5bedb07941ca8d9198f5934d"),
            ObjectId("5bedb07941ca8d9198f5934e")
    ]
}
```

Interrogation de documents

Il arrive que vous ayez besoin de consulter le stock de votre boutique en ligne, pour que les clients puissent voir et acheter ce que vous vendez. Interroger une collection est relativement simple, qu'il s'agisse de tous les documents dans la collection ou uniquement ceux qui répondent à un critère particulier.

Pour interroger des documents, utilisez l'opération `find()`. La commande `find()` possède un seul paramètre de document qui définit les critères à utiliser pour choisir le documents à renvoyer Le

résultat obtenu à partir de `find()` est d'un document formaté en une seule ligne de texte sans sauts de ligne. Pour formater le document de sortie afin d'en faciliter la lecture, utilisez `find().pretty()`. Tous les exemples de cette rubrique utilisent `.pretty()` pour mettre en forme les données de sortie.

Utilisez les quatre documents que vous avez insérés dans le`example`dans les deux exercices précédents —`insertOne()`et`insertMany()`.

Rubriques

- Récupération de tous les documents dans une collection
- Récupération de documents correspondant à une valeur de champ
- Récupération de documents correspondant à un document intégré
- Récupération de documents correspondant à une valeur de champ dans un document intégré
- Récupération de documents correspondant à un tableau
- Récupération de documents correspondant à une valeur dans un tableau
- Récupération de documents à l'aide d'opérateurs

Récupération de tous les documents dans une collection

Pour récupérer tous les documents de votre collection, utilisez l'opération `find()` avec un document de requête vide.

La requête suivante renvoie tous les documents de la collection `example`.

```
db.example.find( {} ).pretty()
```

Récupération de documents correspondant à une valeur de champ

Pour récupérer tous les documents qui correspondent à un champ et à une valeur, utilisez l'opération `find()` avec un document de requête qui identifie les champs et les valeurs à faire correspondre.

En utilisant les documents précédents, cette requête renvoie tous les documents où le champ « Item (Élément) » est défini sur « Pen ».

```
db.example.find( { "Item": "Pen" } ).pretty()
```

Récupération de documents correspondant à un document intégré

Pour rechercher tous les documents qui correspondent à un document intégré, utilisez l'opération `find()` avec un document de requête qui spécifie le nom du document intégré et tous les champs et les valeurs de ce document intégré.

Lorsqu'une correspondance est établie avec un document intégré , le nom de ce document doit être identique à celui spécifié dans la requête. De plus, les champs et les valeurs dans le document intégré doivent correspondre à la requête.

La requête suivante renvoie uniquement le document « Poster Paint ». Cela est dû au fait que « Pen » a des valeurs différentes pour « `OnHand` » et « `MinOnHand` », et que et « Spray Paint » a un champ de plus (`OrderQnty`) que le document de requête.

```
db.example.find({"Inventory": {
    "OnHand": 47,
    "MinOnHand": 50 } } ).pretty()
```

Récupération de documents correspondant à une valeur de champ dans un document intégré

Pour rechercher tous les documents qui correspondent à un document intégré, utilisez l'opération `find()` avec un document de requête qui spécifie le nom du document intégré et tous les champs et les valeurs de ce document intégré.

Au vu des documents précédents, la requête suivante utilise la « notation de points » pour spécifier le document intégré et les champs d'intérêt. Tout document correspondant est renvoyé, quels que soient les autres champs présents dans le document intégré. La requête renvoie « Poster Paint » et « Spray Paint », car ils correspondent tous deux aux champs et aux valeurs spécifiés.

```
db.example.find({"Inventory.OnHand": 47, "Inventory.MinOnHand": 50 }).pretty()
```

Récupération de documents correspondant à un tableau

Pour rechercher tous les documents qui correspondent à un tableau, utilisez l'opération `find()` avec le nom du tableau qui vous intéresse et toutes les valeurs de ce tableau. La requête renvoie tous les documents ayant un tableau avec ce nom et dans lequel les valeurs sont identiques et figurent dans le même ordre que dans la requête.

La requête suivante renvoie uniquement « Pen », car « Poster Paint » a une couleur supplémentaire (blanc) et les couleurs de « Spray Paint » figurent dans un ordre différent.

```
db.example.find( { "Colors": ["Red","Green","Blue","Black"] } ).pretty()
```

Récupération de documents correspondant à une valeur dans un tableau

Pour rechercher tous les documents contenant une valeur de tableau particulière, utilisez l'opération `find()` avec le nom du tableau et la valeur qui vous intéressent.

```
db.example.find( { "Colors": "Red" } ).pretty()
```

L'opération précédente renvoie les trois documents, car chacun d'entre eux possède un tableau nommé `Colors` et la valeur « `Red` » dans le tableau. Si vous spécifiez la valeur « `White` », la requête renvoie uniquement « Poster Paint ».

Récupération de documents à l'aide d'opérateurs

La requête suivante renvoie tous les documents où la valeur « `Inventory.OnHand` » est inférieure à 50.

```
db.example.find(
        { "Inventory.OnHand": { $lt: 50 } } )
```

Pour obtenir une liste des opérateurs de requête pris en charge, consultez Opérateurs de projection et de requête.

Mise à jour des documents

Généralement, vos documents ne sont pas statiques et sont mis à jour dans le cadre de vos flux de travaux applicatifs. Les exemples suivants illustrent les différentes façons de mettre à jour des documents.

Pour mettre à jour un document existant, utilisez l'opération `update()`. L'opération `update()` possède deux paramètres de document. Le premier document identifie le ou les documents à mettre à jour. Le deuxième document spécifie les mises à jour à effectuer.

Lorsque vous mettez à jour un champ existant (que celui-ci soit un champ simple, un tableau ou un document intégré), vous spécifiez le nom du champ et ses valeurs. À la fin de l'opération, c'est comme si le champ de l'ancien document avait été remplacé par le nouveau champ et ses valeurs.

Rubriques

- Mise à jour des valeurs d'un champ existant

- Ajouter un nouveau champ
- Remplacement d'un document intégré
- Insertion de nouveaux champs dans un document intégré
- Suppression d'un champ dans un document
- Suppression d'un champ de plusieurs documents

Mise à jour des valeurs d'un champ existant

Utilisez les quatre documents suivants que vous avez ajoutés précédemment pour les opérations de mise à jour suivantes.

```
{
    "Item": "Ruler",
    "Colors": ["Red","Green","Blue","Clear","Yellow"],
    "Inventory": {
        "OnHand": 47,
        "MinOnHand": 40
    },
    "UnitPrice": 0.89
},
{
    "Item": "Pen",
    "Colors": ["Red","Green","Blue","Black"],
    "Inventory": {
        "OnHand": 244,
        "MinOnHand": 72
    }
},
{
    "Item": "Poster Paint",
    "Colors": ["Red","Green","Blue","Black","White"],
    "Inventory": {
        "OnHand": 47,
        "MinOnHand": 50
    }
},
{
    "Item": "Spray Paint",
    "Colors": ["Black","Red","Green","Blue"],
    "Inventory": {
        "OnHand": 47,
```

```
        "MinOnHand": 50,
        "OrderQnty": 36
    }
}
```

Pour mettre à jour un champ simple

Pour mettre à jour un champ simple, utilisez `update()` avec `$set` pour spécifier le nom du champ et la nouvelle valeur. L'exemple suivant modifie l'élément `Item` de « Pen » en « Gel Pen ».

```
db.example.update(
    { "Item" : "Pen" },
    { $set: { "Item": "Gel Pen" } }
)
```

Les résultats de cette opération ressemblent à ce qui suit.

```
{
    "Item": "Gel Pen",
    "Colors": ["Red","Green","Blue","Black"],
    "Inventory": {
        "OnHand": 244,
        "MinOnHand": 72
    }
}
```

Pour mettre à jour un tableau :

L'exemple suivant remplace le tableau de couleurs existant par un nouveau tableau qui inclut `Orange` et supprime `White` de la liste des couleurs. La nouvelle liste de couleurs est dans l'ordre spécifié dans l'opération `update()`.

```
db.example.update(
    { "Item" : "Poster Paint" },
    { $set: { "Colors": ["Red","Green","Blue","Orange","Black"] } }
)
```

Les résultats de cette opération ressemblent à ce qui suit.

```
{
    "Item": "Poster Paint",
    "Colors": ["Red","Green","Blue","Orange","Black"],
```

```
    "Inventory": {
        "OnHand": 47,
        "MinOnHand": 50
    }
}
```

Ajouter un nouveau champ

Pour modifier un document en ajoutant un ou plusieurs nouveaux champs, utilisez l'opération `update()`, avec un document de requête qui identifie le document à insérer et les nouveaux champs et valeurs à insérer à l'aide de l'opérateur `$set`.

L'exemple suivant ajoute le champ `UnitPrice` avec la valeur `3.99` pour le document Spray Paints. Notez que la valeur `3.99` est numérique et non une chaîne.

```
db.example.update(
    { "Item": "Spray Paint" },
    { $set: { "UnitPrice": 3.99 } }
)
```

Les résultats de cette opération ressemblent à ce qui suit (format JSON).

```
{
    "Item": "Spray Paint",
    "Colors": ["Black","Red","Green","Blue"],
    "Inventory": {
        "OnHand": 47,
        "MinOnHand": 50,
        "OrderQnty": 36
    },
    "UnitPrice": 3.99
}
```

Remplacement d'un document intégré

Pour modifier un document en remplaçant un document intégré, utilisez l'opération `update()` avec des documents qui identifient le document intégré et ses nouveaux champs et valeurs à insérer à l'aide de l'opérateur `$set`.

Prenons l'exemple du document suivant :

```
db.example.insert({
```

```
    "DocName": "Document 1",
    "Date": {
        "Year": 1987,
        "Month": 4,
        "Day": 18
    }
})
```

Pour remplacer un document intégré

L'exemple suivant remplace le document actuel Date par un nouveau, qui dispose uniquement des champs `Month` et `Day`, `Year` ayant été supprimé.

```
db.example.update(
    { "DocName" : "Document 1" },
    { $set: { "Date": { "Month": 4, "Day": 18 } } }
)
```

Les résultats de cette opération ressemblent à ce qui suit.

```
{
    "DocName": "Document 1",
    "Date": {
        "Month": 4,
        "Day": 18
    }
}
```

Insertion de nouveaux champs dans un document intégré

Pour ajouter des champs à un document intégré

Pour modifier un document en ajoutant un ou plusieurs nouveaux champs à un document intégré, utilisez l'opération `update()`, avec des documents qui identifient le document intégré et la « notation de points » pour spécifier le document intégré et les nouveaux champs et valeurs à insérer à l'aide de l'opérateur `$set`.

Au vu du document suivant, le code ci-après utilise la « notation de points » pour insérer les champs `Year` et `DoW` dans le document intégré `Date` et `Words` dans le document parent.

```
{
```

```
    "DocName": "Document 1",
    "Date": {
        "Month": 4,
        "Day": 18
    }
}
```

```
db.example.update(
    { "DocName" : "Document 1" },
    { $set: { "Date.Year": 1987,
              "Date.DoW": "Saturday",
              "Words": 2482 } }
)
```

Les résultats de cette opération ressemblent à ce qui suit.

```
{
    "DocName": "Document 1",
    "Date": {
        "Month": 4,
        "Day": 18,
        "Year": 1987,
        "DoW": "Saturday"
    },
    "Words": 2482
}
```

Suppression d'un champ dans un document

Pour modifier un document en supprimant un champ dans le document, utilisez l'opération `update()`, avec un document de requête qui identifie le document dans lequel il faut supprimer le champ, ainsi que le champ à supprimer à l'aide de l'opérateur `$unset`.

L'exemple suivant supprime le champ `Words` dans le document précédent.

```
db.example.update(
    { "DocName" : "Document 1" },
    { $unset: { Words:1 } }
)
```

Les résultats de cette opération ressemblent à ce qui suit.

```
{
    "DocName": "Document 1",
    "Date": {
        "Month": 4,
        "Day": 18,
        "Year": 1987,
        "DoW": "Saturday"
    }
}
```

Suppression d'un champ de plusieurs documents

Pour modifier un document en supprimant un champ de plusieurs documents, utilisez l'opération `update()` avec l'opérateur `$unset` et l'option `multi` définie sur `true`.

L'exemple suivant supprime le champ `Inventory` de tous les documents de l’exemple de collection. Si un document n'a pas de champ `Inventory`, aucune action n'est effectuée sur celui-ci. Si `multi: true` n'est pas spécifié, l'action est uniquement effectuée sur le premier document qui répond aux critères.

```
db.example.update(
    {},
    { $unset: { Inventory:1 } },
    { multi: true }
)
```

Suppression de documents

Pour supprimer un document de votre base de données, utilisez l'opération `remove()`, en indiquant quel document il faut supprimer. Le code suivant supprime « Gel Pen » de votre collection `example`.

```
db.example.remove( { "Item": "Gel Pen" } )
```

Pour supprimer tous les documents de votre base de données, utilisez l'opération `remove()` avec une requête vide, comme dans l'exemple suivant.

```
db.example.remove( { } )
```

Commencez à utiliser Amazon DocumentDB

Il existe de nombreuses manières de se connecter et de démarrer avec Amazon DocumentDB. Nous avons créé ce guide parce que nous avons trouvé que c'était le moyen le plus rapide, le plus simple et le plus simple pour les utilisateurs de commencer à utiliser notre puissante base de données de documents. Ce guide utilise AWS Cloud9un terminal Web pour connecter et interroger votre cluster Amazon DocumentDB à l'aide du shell mongo directement depuis le. AWS Management Console Les nouveaux clients éligibles au niveau AWS gratuit peuvent utiliser Amazon DocumentDB AWS Cloud9 gratuitement. Si votre AWS Cloud9 environnement ou votre cluster Amazon DocumentDB utilise des ressources au-delà du niveau gratuit, les AWS tarifs normaux vous sont facturés pour ces ressources. Ce guide vous permettra de démarrer avec Amazon DocumentDB en moins de 15 minutes.

> **Note**
>
> Les instructions de ce guide concernent spécifiquement la création et la connexion à des clusters basés sur des instances Amazon DocumentDB. Si vous souhaitez créer des clusters élastiques Amazon DocumentDB et vous y connecter, consultez. Démarrez avec les clusters élastiques Amazon DocumentDB

Rubriques

Si vous préférez vous connecter à votre Amazon DocumentDB depuis votre machine locale en créant une connexion SSH vers une instance Amazon EC2, consultez les instructions Connect with EC2

Prérequis

Avant de créer votre premier cluster Amazon DocumentDB, vous devez effectuer les opérations suivantes :

Créez un compte Amazon Web Services (AWS)

Avant de pouvoir commencer à utiliser Amazon DocumentDB, vous devez disposer d'un compte Amazon Web Services (AWS). Le AWS compte est gratuit. Vous payez uniquement les services et les ressources que vous utilisez.

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

Pour vous inscrire à un Compte AWS

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

   Dans le cadre de la procédure d‘inscription, vous recevrez un appel téléphonique 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 en matière de sécurité consiste à attribuer un accès administratif à un utilisateur et à n'utiliser que l'utilisateur root pour effectuer les tâches nécessitant un accès utilisateur root.

Configurez les autorisations AWS Identity and Access Management (IAM) nécessaires.

L'accès à la gestion des ressources Amazon DocumentDB telles que les clusters, les instances et les groupes de paramètres de cluster nécessite des informations d'identification AWS pouvant être utilisées pour authentifier vos demandes. Pour plus d’informations, consultez Identity and Access Management pour Amazon DocumentDB.

1. Dans la barre de recherche du AWS Management Console, tapez IAM et sélectionnez IAM dans le menu déroulant qui apparaît.
2. Une fois dans la console IAM, sélectionnez Utilisateurs dans le volet de navigation.
3. Sélectionnez votre nom d'utilisateur.

4. Cliquez sur le bouton Ajouter des autorisations.
5. Sélectionnez Attach existing policies directly (Attacher directement les politiques existantes).
6. Tapez `AmazonDocDBFullAccess` dans la barre de recherche et sélectionnez-la une fois qu'elle apparaît dans les résultats de recherche.
7. Cliquez sur le bouton bleu en bas qui indique Suivant : Réviser.
8. Cliquez sur le bouton bleu en bas qui indique Ajouter des autorisations.

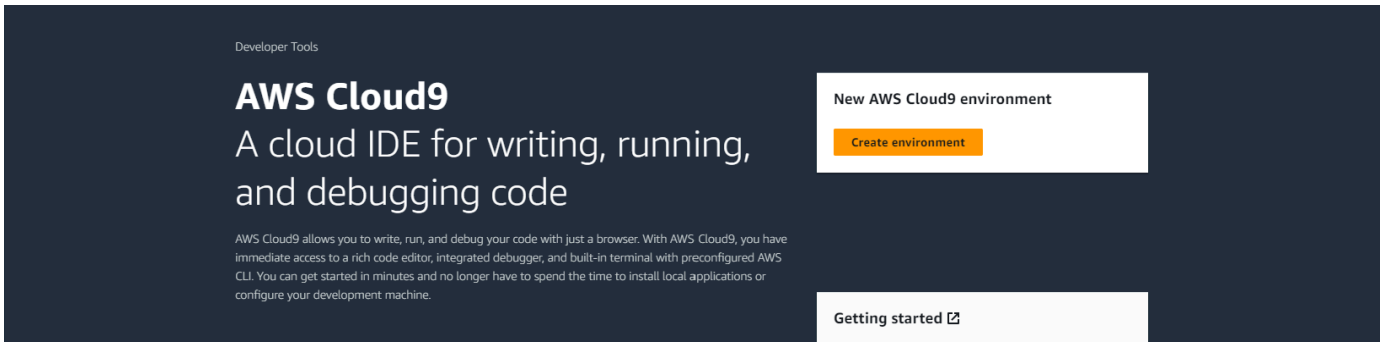
Création d'un Amazon Virtual Private Cloud (Amazon VPC)

Cette étape n'est nécessaire que si vous ne possédez pas encore d'Amazon VPC par défaut. Si ce n'est pas le cas, suivez l'étape 1 de la section Getting Started with Amazon VPC User Guide. Cela prendra moins de cinq minutes.

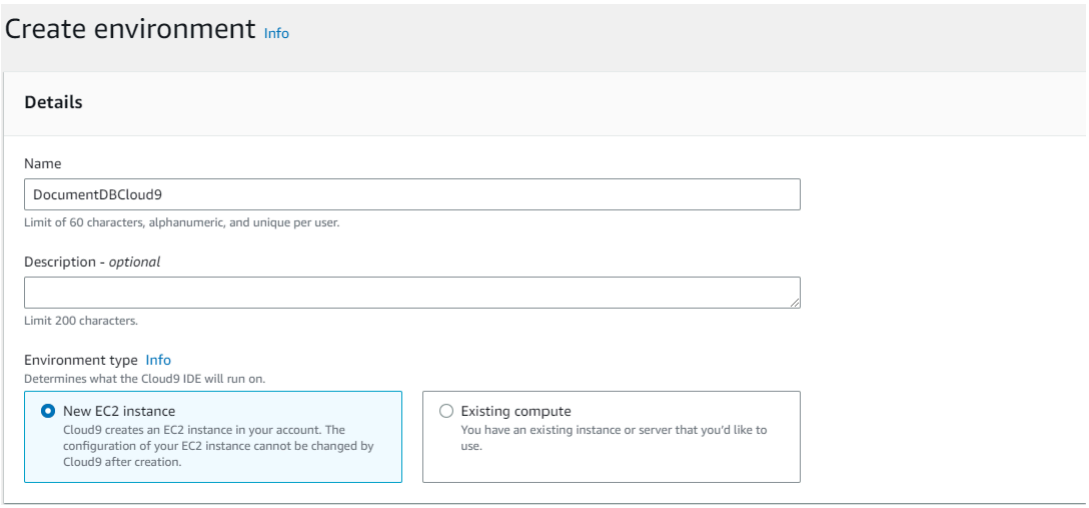
Étape 1 : Création d'un AWS Cloud9 environnement

AWS Cloud9 fournit un terminal Web que vous pouvez utiliser pour vous connecter à votre cluster Amazon DocumentDB et l'interroger à l'aide du shell mongo.

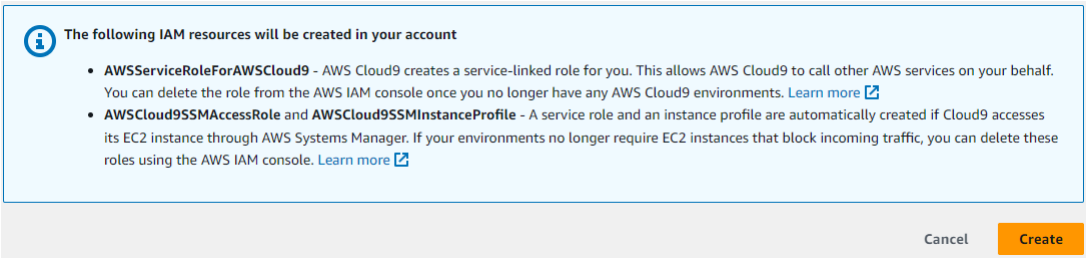
1. AWS Management Console Accédez à la AWS Cloud9 console et choisissez Create environment.



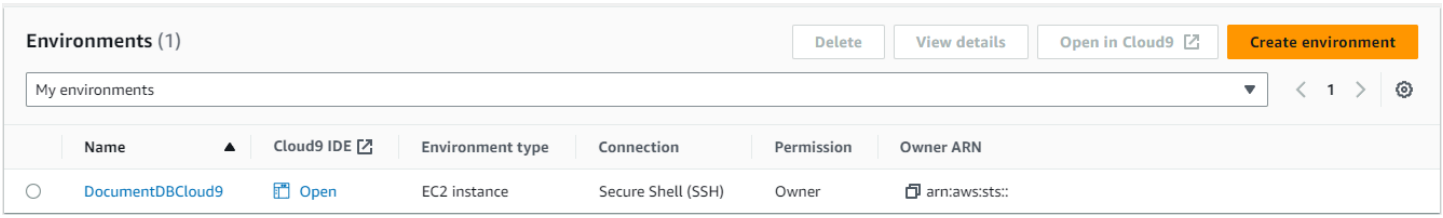
2. Dans la section Détails de la boîte de dialogue Créer un environnement, entrez `DocumentDBCloud9` dans le champ Nom.



3. Pour les sections Nouvelle instance EC2, Paramètres réseau et Tags, laissez le paramètre par défaut tel quel et cliquez sur Créer en bas de l'écran.



Votre nouvel AWS Cloud9 environnement apparaît dans le tableau Environnements :

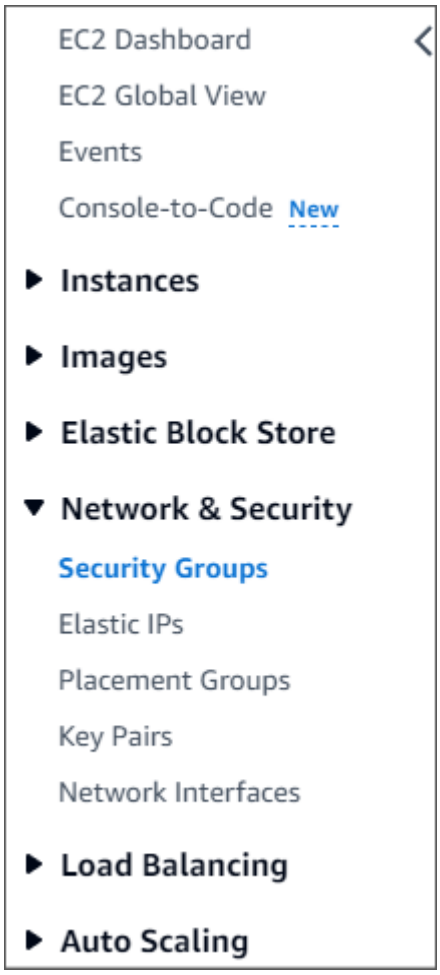


> **Note**
>
> Le provisionnement de l' AWS Cloud9 environnement peut prendre jusqu'à trois minutes.

Étape 2 : Créer un groupe de sécurité

Ce groupe de sécurité vous permettra de vous connecter à votre cluster Amazon DocumentDB depuis votre AWS Cloud9 environnement.

1. Sur la console de gestion Amazon EC2, sous Réseau et sécurité, sélectionnez Groupes de sécurité.



2. Sélectionnez Create security group (Créer un groupe de sécurité).

3. Dans la section Informations de base :

   a. Sous Security group name (Nom du groupe de sécurité), saisissez `demoDocDB`.

   b. Pour Description, entrez une description.

   c. Pour le VPC, acceptez l'utilisation de votre VPC par défaut.

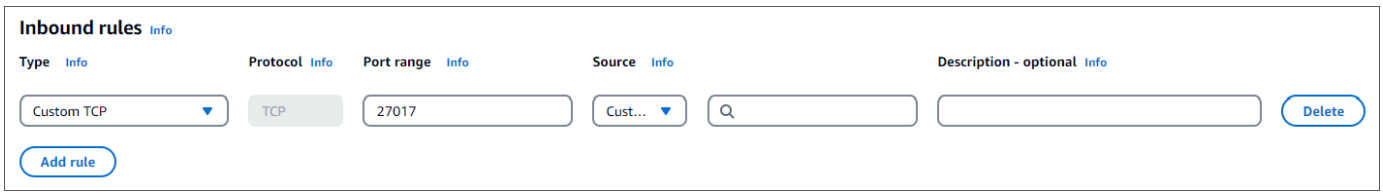
4. Dans la section Règles entrantes, choisissez Ajouter une règle.

   a. Pour Type, choisissez Règle TCP personnalisée.

   b. Dans Portée de ports, entrez`27017`.

   c. Pour Source, choisissez le groupe de sécurité pour l' AWS Cloud9 environnement que vous venez de créer. Pour voir la liste des groupes de sécurité disponibles, entrez `cloud9` dans le champ de recherche situé à droite du champ Source. Choisissez le groupe de sécurité portant le nom `aws-cloud9-<environment name>`.

   d. Pour Destination, choisissez Personnalisé. Dans le champ à côté, recherchez le groupe de sécurité que vous venez d'appelerdemoEC2. Vous devrez peut-être actualiser votre navigateur pour que la console Amazon EC2 renseigne automatiquement le nom de la source. `demoEC2`



> **Note**
>
> Le port 27017 est le port par défaut pour Amazon DocumentDB.

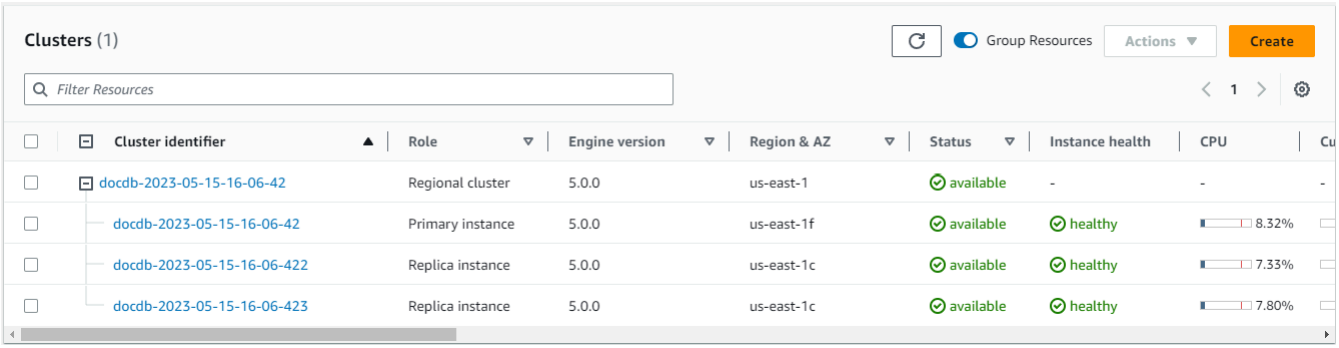
5. Acceptez toutes les autres valeurs par défaut et choisissez Create security group.

Étape 3 : créer un cluster Amazon DocumentDB

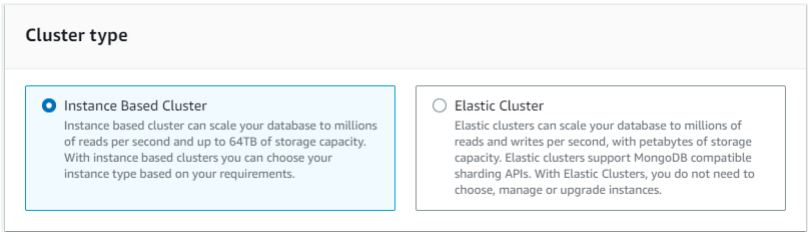
Au cours de cette étape, vous allez créer un cluster Amazon DocumentDB en utilisant le groupe de sécurité que vous avez créé à l'étape précédente.

> **Note**
>
> Les instructions de cette étape concernent spécifiquement la création de clusters basés sur des instances Amazon DocumentDB. Si vous souhaitez créer des clusters élastiques Amazon DocumentDB, consultez. Démarrez avec les clusters élastiques Amazon DocumentDB

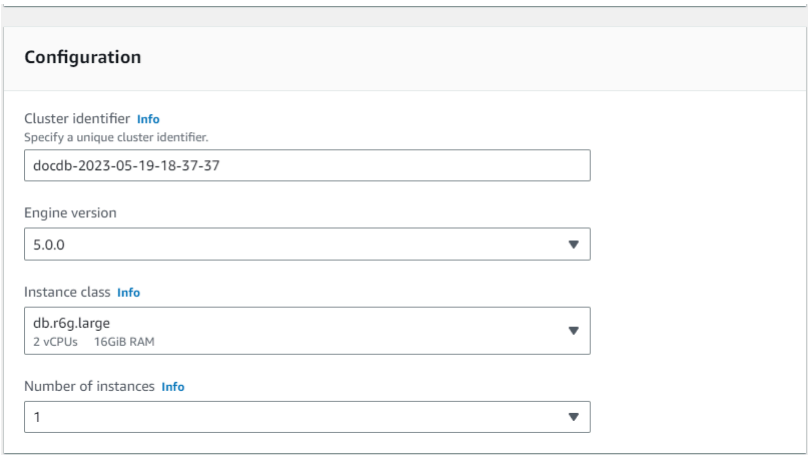
1. Sur la console de gestion Amazon DocumentDB, sous Clusters, choisissez Create.



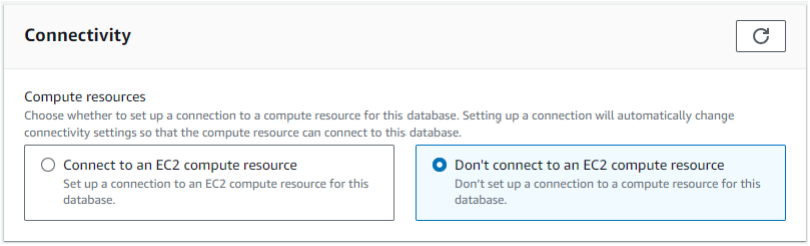
2. Sur la page Créer un cluster Amazon DocumentDB, dans la section Type de cluster, sélectionnez Instance Based Clusters (il s'agit de l'option par défaut).



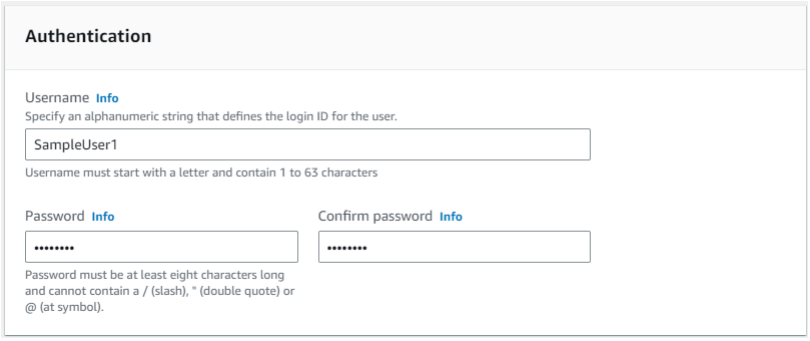
3. Dans la section Configuration, sélectionnez 1 instance. Le choix d'une instance permet de minimiser les coûts. S'il s'agissait d'un système de production, nous vous recommandons de fournir trois instances pour une haute disponibilité. Vous pouvez conserver les valeurs par défaut des autres paramètres de la section Configuration.



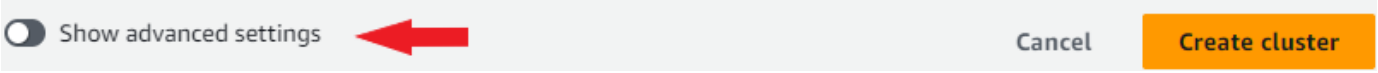
4. Pour Connectivité, conservez le paramètre par défaut Ne pas se connecter à une ressource de calcul EC2.



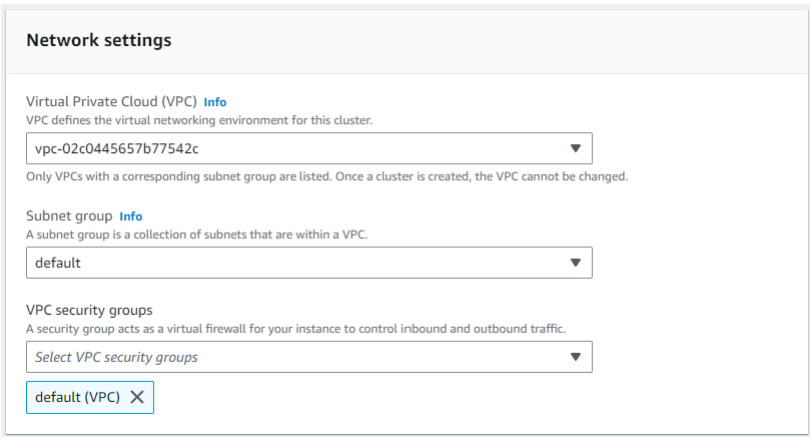
5. Dans la section Authentification, entrez les informations de connexion.



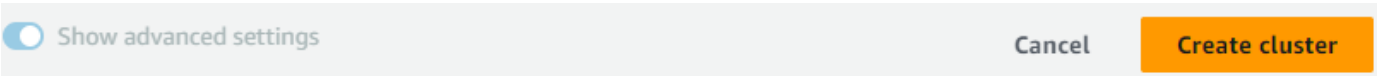
6. Activez Afficher les paramètres avancés.



7. Dans la section Paramètres réseau, pour les groupes de sécurité VPC, choisissez DemoDocDB (VPC) si vous créez un cluster de test ou de démonstration. Si vous créez un cluster pour un système de production, choisissez par défaut (VPC) ou si vous souhaitez créer un groupe de sécurité VPC spécifique, consultez la section Groupes de sécurité dans le guide de l'utilisateur d'Amazon Virtual Private Cloud.



8. Choisissez Créer un cluster.



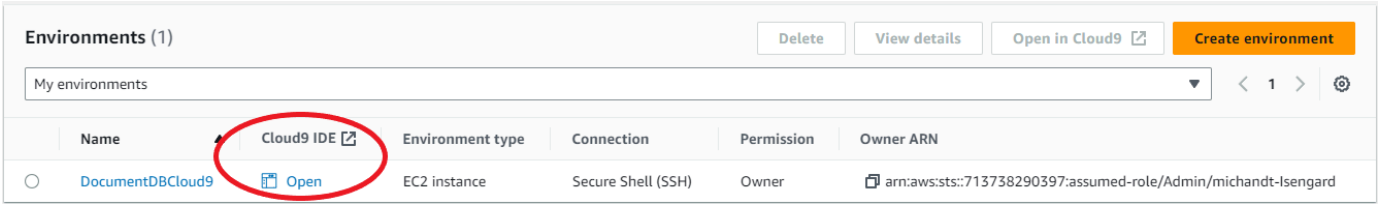
Amazon DocumentDB est en train de provisionner votre cluster, ce qui peut prendre jusqu'à quelques minutes. Vous pouvez vous connecter à votre cluster lorsque le statut du cluster et de l'instance s'affichent sous la forme**`available`**.

> **Note**
>
> Pour plus d'informations sur les valeurs d'état du cluster, consultez Valeurs de statut de cluster le chapitre Monitoring Amazon DocumentDB.
> Pour plus d'informations sur les valeurs de statut des instances, consultez Valeurs de l'état d'instance le chapitre Monitoring Amazon DocumentDB.

Étape 4 : Installation du shell Mongo

Vous allez maintenant installer le shell mongo dans AWS Cloud9 l'environnement que vous avez créé à l'étape 1. Le shell mongo est un utilitaire de ligne de commande que vous utilisez pour connecter et interroger votre cluster Amazon DocumentDB.

1. Si votre AWS Cloud9 environnement est toujours ouvert après l'étape 1, revenez à cet environnement et passez à l'instruction 3. Si vous avez quitté votre AWS Cloud9 environnement, dans la console de AWS Cloud9 gestion, sous Environnements, recherchez l'environnement intitulé DocumentDBCloud9. Choisissez Ouvrir dans la colonne Cloud9 IDE.



2. À l'invite de commande, créez le fichier de référentiel à l'aide de la commande suivante :

```
echo -e "[mongodb-org-4.0] \nname=MongoDB Repository\nbaseurl=https://
repo.mongodb.org/yum/amazon/2013.03/mongodb-org/4.0/x86_64/\ngpgcheck=1 \nenabled=1
 \ngpgkey=https://www.mongodb.org/static/pgp/server-4.0.asc" | sudo tee /etc/
yum.repos.d/mongodb-org-4.0.repo
```

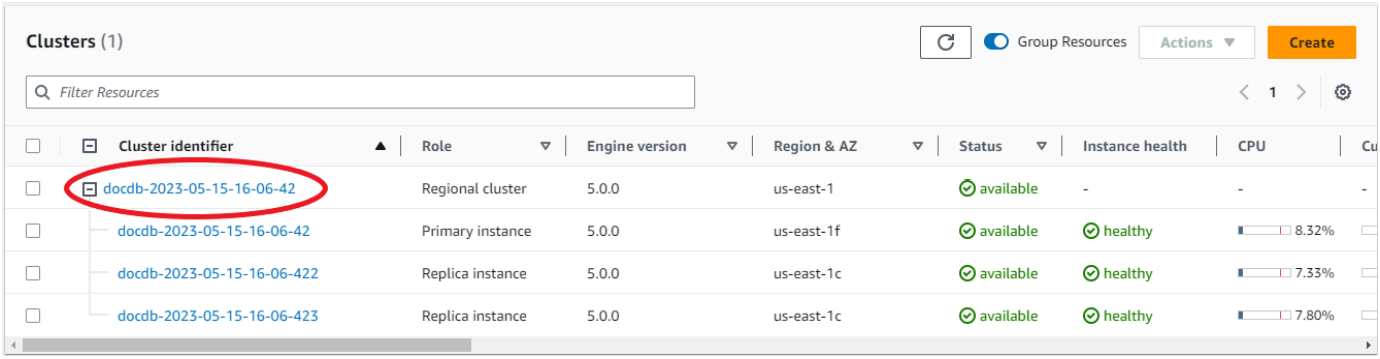
3. Une fois l'opération terminée, installez le shell mongo avec la commande suivante :

```
sudo yum install -y mongodb-org-shell
```

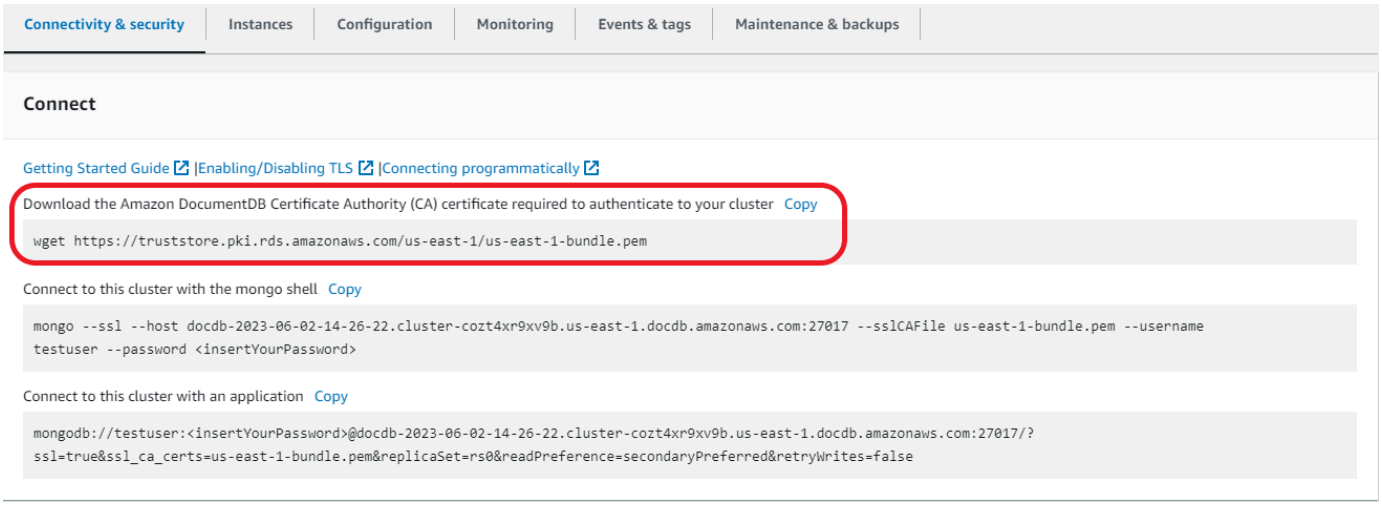
Étape 5 : Connectez-vous à votre cluster Amazon DocumentDB

Vous allez maintenant vous connecter à votre cluster Amazon DocumentDB à l'aide du shell mongo que vous avez installé à l'étape 4.

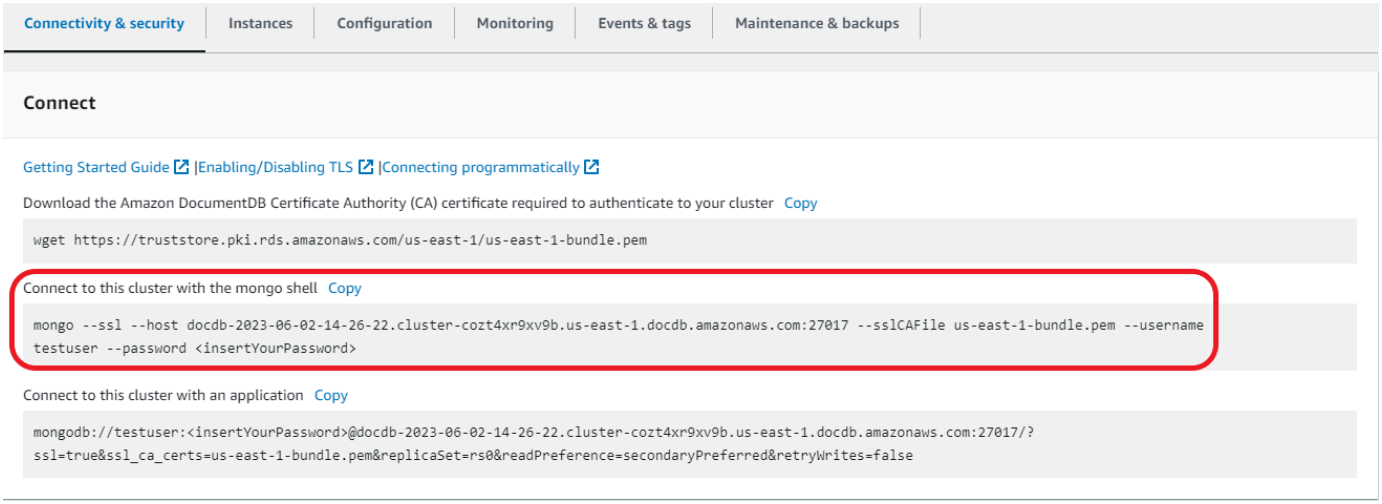
1. Sur la console de gestion Amazon DocumentDB, sous Clusters, localisez votre cluster. Choisissez le cluster que vous avez créé en cliquant sur son identifiant.



2. E ncryption-in-transit est activé par défaut sur Amazon DocumentDB. Vous pouvez éventuellement désactiver le protocole TLS. Pour télécharger le certificat actuel requis pour vous authentifier auprès de votre cluster, dans l'onglet Connectivité et sécurité, dans la section Connect, sous Télécharger le certificat Amazon DocumentDB Certificate Authority (CA) requis pour l'authentification auprès de votre cluster, copiez la chaîne de connexion fournie. Retournez dans votre AWS Cloud9 environnement et collez la chaîne de connexion.



3. Revenez à votre cluster dans la console Amazon DocumentDB, sous l'onglet Connectivité et sécurité, dans la section Connect, sous Connect to this cluster with the mongo shell, copiez la chaîne de connexion fournie. Omettez de copier `<insertYourPassword>` afin que le shell mongo vous demande le mot de passe lorsque vous vous connectez.



   Retournez dans votre AWS Cloud9 environnement et collez la chaîne de connexion.

Lorsque vous entrez votre mot de passe et que votre invite devient `rs0:PRIMARY>` une invite, vous êtes connecté avec succès à votre cluster Amazon DocumentDB.

> **Note**
>
> Pour plus d'informations sur le dépannage, consultez la section Résolution des problèmes liés à Amazon DocumentDB.

Étape 6 : Insérer et interroger des données

Maintenant que vous êtes connecté à votre cluster, vous pouvez exécuter quelques requêtes pour vous familiariser avec l'utilisation d'une base de données de documents.

1. Pour insérer un seul document, entrez les informations suivantes :

```
db.collection.insert({"hello":"DocumentDB"})
```

2. Vous obtenez le résultat suivant :

```
WriteResult({ "nInserted" : 1 })
```

3. Vous pouvez lire le document que vous avez écrit avec la `findOne()` commande (car il ne renvoie qu'un seul document). Entrez les informations suivantes :

```
db.collection.findOne()
```

4. Vous obtenez le résultat suivant :

```
{ "_id" : ObjectId("5e401fe56056fda7321fbd67"), "hello" : "DocumentDB"
                        }
```

5. Pour effectuer quelques requêtes supplémentaires, considérez un cas d'utilisation de profils de jeu. Tout d'abord, insérez quelques entrées dans une collection intituléeprofiles. Entrez les informations suivantes :

```
db.profiles.insertMany([
                    { "_id" : 1, "name" : "Matt", "status": "active", "level": 12,
 "score":202},
                    { "_id" : 2, "name" : "Frank", "status": "inactive", "level":
 2, "score":9},
                    { "_id" : 3, "name" : "Karen", "status": "active", "level": 7,
 "score":87},
                    { "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
 "score":27}
                    ])
```

6. Vous obtenez le résultat suivant :

```
{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4 ] }
```

7. Utilisez la `find()` commande pour renvoyer tous les documents de la collection de profils. Entrez les informations suivantes :

```
db.profiles.find()
```

8. Vous obtiendrez une sortie qui correspondra aux données que vous avez saisies à l'étape 5.
9. Utilisez une requête pour un seul document à l'aide d'un filtre. Entrez les informations suivantes :

```
db.profiles.find({name: "Katie"})
```

10. Vous devriez récupérer cette sortie :

```
{ "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
                        "score":27}
```

11. Essayons maintenant de trouver un profil et de le modifier à l'aide de la `findAndModify` commande. Nous allons donner dix points supplémentaires à l'utilisateur Matt avec le code suivant :

```
db.profiles.findAndModify({
                query: { name: "Matt", status: "active"},
                update: { $inc: { score: 10 } }
            })
```

12. Vous obtenez le résultat suivant (notez que son score n'a pas encore augmenté) :

```
{
        "_id" : 1,
        "name" : "Matt",
        "status" : "active",
        "level" : 12,
        "score" : 202
}
```

13. Vous pouvez vérifier que son score a changé avec la requête suivante :

```
db.profiles.find({name: "Matt"})
```

14. Vous obtenez le résultat suivant :

```
{ "_id" : 1, "name" : "Matt", "status" : "active", "level" : 12, "score"
```

```
                    : 212 }
```

Étape 7 : Explorez

Félicitations ! Vous avez terminé avec succès le guide de démarrage d'Amazon DocumentDB.

Quelle est la prochaine étape ? Découvrez comment tirer pleinement parti de cette base de données grâce à certaines de ses fonctionnalités les plus populaires :

- Gestion d'Amazon DocumentDB
- Dimensionnement
- Sauvegarde et restauration

> **Note**
>
> Le cluster que vous avez créé à partir de cet exercice de démarrage continuera à générer des coûts à moins que vous ne le supprimiez. Pour obtenir des instructions, consultez Supprimer un cluster Amazon DocumentDB.

Démarrage rapide avec Amazon DocumentDB AWS CloudFormation

Cette section contient des étapes et d'autres informations pour vous aider à démarrer rapidement avec Amazon DocumentDB (compatible avec MongoDB) en utilisant. AWS CloudFormation Pour obtenir des informations générales sur Amazon DocumentDB, consultez. Qu'est-ce qu'Amazon DocumentDB (avec compatibilité avec MongoDB)

Ces instructions utilisent un AWS CloudFormation modèle pour créer un cluster et des instances dans votre Amazon VPC par défaut. Pour obtenir des instructions afin de créer ces ressources vous-même, consultez Commencez à utiliser Amazon DocumentDB.

> **Important**
>
> La AWS CloudFormation pile créée par ce modèle crée plusieurs ressources, notamment des ressources dans Amazon DocumentDB (par exemple, un cluster et des instances) et Amazon Elastic Compute Cloud (par exemple, un groupe de sous-réseaux).
> Certaines de ces ressources ne sont pas couvertes par une offre gratuite. Pour obtenir des informations sur les tarifs, consultez les sections Tarification Amazon DocumentDB et Tarification Amazon EC2. Vous pouvez supprimer la pile lorsque vous n'en avez plus besoin pour arrêter les frais.

Cette AWS CloudFormation pile est destinée uniquement à des fins de didacticiel. Si vous utilisez ce modèle pour un environnement de production, nous vous recommandons d'utiliser des politiques et des mesures de sécurité IAM plus strictes. Pour plus d'informations sur la sécurisation des ressources, consultez Amazon VPC Security et Amazon EC2 Network and Security.

Rubriques

Prérequis

Avant de créer un cluster Amazon DocumentDB, vous devez disposer des éléments suivants :

- Un Amazon VPC par défaut
- Les autorisations IAM nécessaires

Autorisations IAM nécessaires

Les autorisations suivantes vous permettent de créer des ressources pour la pile AWS CloudFormation :

AWS Politiques gérées

- `AWSCloudFormationReadOnlyAccess`
- `AmazonDocDBFullAccess`

Autorisations IAM supplémentaires

La politique suivante décrit les autorisations supplémentaires requises pour créer et supprimer cette AWS CloudFormation pile.

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
            "Effect": "Allow",
            "Action": [
                "iam:GetSSHPublicKey",
                "iam:ListSSHPublicKeys",
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:PutRolePolicy",
                "iam:CreateInstanceProfile",
                "iam:AddRoleToInstanceProfile",
                "iam:GetAccountSummary",
                "iam:ListAccountAliases",
                "iam:GetRole",
                "iam:DeleteRole",
```

```
                "iam:RemoveRoleFromInstanceProfile",
                "iam:DeleteRolePolicy",
                "iam:DeleteInstanceProfile",
                "cloudformation:*Stack",
                "ec2:DescribeKeyPairs",
                "ec2:*Vpc",
                "ec2:DescribeInternetGateways",
                "ec2:*InternetGateway",
                "ec2:createTags",
                "ec2:*VpcAttribute",
                "ec2:DescribeRouteTables",
                "ec2:*RouteTable",
                "ec2:*Subnet",
                "ec2:*SecurityGroup",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:DescribeVpcEndpoints",
                "ec2:*VpcEndpoint",
                "ec2:*SubnetAttribute",
                "ec2:*Route",
                "ec2:*Instances",
                "ec2:DeleteVpcEndpoints"
            ],
            "Resource": "*"
        },
        {
            "Sid": "iamPassRole",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "rds.amazonaws.com"
                }
            }
        }
    ]
}
```

> **Note**
>
> Les autorisations en caractères gras dans la stratégie précédente sont uniquement requises pour supprimer une pile : `iam:DeleteRole`, `iam:RemoveRoleFromInstanceProfile`, `iam:DeleteRolePolicy`, `iam:DeleteInstanceProfile`, et

`ec2:DeleteVpcEndpoints`. Notez également que `ec2:*Vpc` accorde des autorisations `ec2:DeleteVpc`.

Paire de clés Amazon EC2

Vous devez disposer d'une paire de clés (et du fichier PEM) dans la région où vous allez créer la AWS CloudFormation pile. Si vous devez créer une paire de clés, consultez la section Création d'une paire de clés à l'aide d'Amazon EC2 dans le guide de l'utilisateur Amazon EC2.

Lancement d'une pile Amazon DocumentDB AWS CloudFormation

Cette section explique comment lancer et configurer une pile Amazon DocumentDB. AWS CloudFormation

1. Connectez-vous à l' AWS Management Console adressehttps://console.aws.amazon.com/.
2. Le tableau suivant répertorie les modèles de pile Amazon DocumentDB pour chacun d'entre eux. Région AWS Choisissez Launch Stack dans lequel Région AWS vous souhaitez lancer votre stack.

Région	Afficher le modèle	Afficher dans Designer	Lancer
USA Est (Ohio)	Afficher le modèle	Afficher dans Designer	Launch Stack
USA Est (Virginie du Nord)	Afficher le modèle	Afficher dans Designer	Launch Stack
USA Ouest (Oregon)	Afficher le modèle	Afficher dans Designer	Launch Stack
Asie-Pacifique (Mumbai)	Afficher le modèle	Afficher dans Designer	Launch Stack
Asie-Pacifique (Séoul)	Afficher le modèle	Afficher dans Designer	Launch Stack

Région	Afficher le modèle	Afficher dans Designer	Lancer
Asie-Pacifique (Singapour)	Afficher le modèle	Afficher dans Designer	
Asie-Pacifique (Sydney)	Afficher le modèle	Afficher dans Designer	
Asie-Pacifique (Tokyo)	Afficher le modèle	Afficher dans Designer	
Canada (Centre)	Afficher le modèle	Afficher dans Designer	
Europe (Francfort)	Afficher le modèle	Afficher dans Designer	
Europe (Irlande)	Afficher le modèle	Afficher dans Designer	
Europe (Londres)	Afficher le modèle	Afficher dans Designer	
Europe (Paris)	Afficher le modèle	Afficher dans Designer	

3. Create stack — Décrit le modèle Amazon DocumentDB que vous avez sélectionné. Chaque pile est basée sur un modèle (un fichier JSON ou YAML) qui contient la configuration AWS des ressources que vous souhaitez inclure dans la pile. Comme vous avez choisi de lancer une pile à partir des modèles fournis ci-dessus, votre modèle a déjà été configuré pour créer une pile Amazon DocumentDB pour le modèle Région AWS que vous avez choisi.

   Lorsque vous lancez une AWS CloudFormation pile, la protection contre la suppression de votre cluster Amazon DocumentDB est désactivée par défaut. Si vous souhaitez activer la protection contre la suppression pour votre cluster, procédez comme suit. Sinon, choisissez Suivant pour passer à l'étape suivante.

   Pour activer la protection contre la suppression pour votre cluster Amazon DocumentDB :

1. Choisissez Afficher dans le concepteur dans le coin inférieur droit de la page Créer une pile.
2. Modifiez le modèle à l'aide des éditeurs JSON et YAML intégrés dans la page AWS CloudFormation Designer de la console qui en résulte. Faites défiler jusqu'à la section `Resources` et modifiez-la pour inclure `DeletionProtection`, comme suit. Pour plus d'informations sur l'utilisation de AWS CloudFormation Designer, voir Qu'est-ce que AWS CloudFormation Designer ? .

   JSON:

```
"Resources": {
    "DBCluster": {
        "Type": "AWS::DocDB::DBCluster",
        "DeletionPolicy": "Delete",
        "Properties": {
            "DBClusterIdentifier": {
                "Ref": "DBClusterName"
            },
            "MasterUsername": {
                "Ref": "MasterUser"
            },
            "MasterUserPassword": {
                "Ref": "MasterPassword"
            },
            "DeletionProtection": "true"
        }
    },
```

   YAML :

```
Resources:
  DBCluster:
    Type: 'AWS::DocDB::DBCluster'
    DeletionPolicy: Delete
    Properties:
      DBClusterIdentifier: !Ref DBClusterName
      MasterUsername: !Ref MasterUser
      MasterUserPassword: !Ref MasterPassword
      DeletionProtection: 'true'
```

3. Choisissez Créer une pile (

)

dans le coin supérieur gauche de la page pour enregistrer vos modifications et créer une pile avec ces modifications activées.

4. Après avoir enregistré vos modifications, vous serez redirigé vers la page Créer une pile.
5. Choisissez Next (Suivant) pour continuer.

4. Spécifiez les détails de la pile : entrez le nom de la pile et les paramètres de votre modèle. Les paramètres sont définis dans votre modèle et vous permettent d'entrer des valeurs personnalisées lorsque vous créez ou mettez à jour une pile.

- Sous Nom de la pile, entrez un nom pour votre pile ou acceptez le nom fourni. Le nom de la pile peut inclure des lettres (A—Z et a—z), des chiffres (0—9) et des tirets (—).
- Sous Paramètres, entrez les détails suivants :

  - DB ClusterName — Entrez le nom de votre cluster Amazon DocumentDB ou acceptez le nom fourni.

    Contraintes d'attribution de nom relatives à un cluster :

    - La longueur est de [1 à 63] lettres, chiffres ou traits d'union.
    - Le premier caractère doit être une lettre.
    - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
    - Doit être unique pour tous les clusters d'Amazon RDS, Neptune et Amazon Compte AWS DocumentDB par région.
  - DB InstanceClass — Dans la liste déroulante, sélectionnez la classe d'instance pour votre cluster Amazon DocumentDB.
  - DB InstanceName — Entrez un nom pour votre instance Amazon DocumentDB ou acceptez le nom fourni.

    Contraintes d'affectation de noms :

    - La longueur est de [1 à 63] lettres, chiffres ou traits d'union.
    - Le premier caractère doit être une lettre.
    - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
    - Doit être unique pour toutes les instances d'Amazon RDS, Neptune et Amazon Compte AWS DocumentDB par région.
  - MasterPassword — Le mot de passe du compte administrateur de la base de données.

- MasterUser— Nom d'utilisateur du compte administrateur de la base de données. Le MasterUser doit commencer par une lettre et ne peut contenir que des caractères alphanumériques.

Choisissez Suivant pour enregistrer vos modifications et continuer.

5. Configurer les options de la pile : configurez les balises, les autorisations et les options supplémentaires de votre pile.

   - Balises — Spécifiez les paires de balises (clé-valeur) à appliquer aux ressources de votre pile. Vous pouvez ajouter jusqu'à 50 balises uniques pour chaque pile.
   - Autorisations : facultatif. Choisissez un rôle IAM pour définir explicitement AWS CloudFormation comment créer, modifier ou supprimer des ressources dans la pile. Si vous ne choisissez aucun rôle, AWS CloudFormation utilise les autorisations en fonction de vos informations d'identification utilisateur. Avant de spécifier un rôle de service, vérifiez que vous êtes autorisé à le transmettre (`iam:PassRole`). L'autorisation `iam:PassRole` indique les rôles que vous pouvez utiliser.

     > **Note**
     >
     > Lorsque vous spécifiez un rôle de service, utilisez AWS CloudFormation toujours ce rôle pour toutes les opérations effectuées sur cette pile. Les autres utilisateurs autorisés à effectuer des opérations sur cette pile peuvent utiliser ce rôle, même s'ils ne sont pas autorisés à le transmettre. Si le rôle comprend des autorisations que l'utilisateur ne devrait pas avoir, vous avez peut-être remonté accidentellement ses autorisations. Vérifiez que le rôle accorde les privilèges les plus faibles.

   - Options avancées : vous pouvez définir les options avancées suivantes :

     - Stack policy — Facultatif. Définit les ressources que vous voulez protéger contre les mises à jour accidentelles pendant une mise à jour de pile. Par défaut, toutes les ressources peuvent être mises à jour pendant la mise à jour d'une pile.

       Vous pouvez entrer la politique de pile directement au format JSON ou charger un fichier JSON contenant la politique de pile. Pour de plus amples informations, veuillez consulter Éviter les mises à jour des ressources de la pile.
     - Configuration du rollback — Facultatif. Spécifiez CloudWatch les alarmes Logs AWS CloudFormation à surveiller lors de la création et de la mise à jour de la pile. Si l'opération dépasse un seuil d'alarme, AWS CloudFormation annulez-le.

- Options de notification : facultatif. Spécifiez des rubriques pour Simple Notification System (SNS).
- Options de création de piles : facultatif. Vous pouvez spécifier les options suivantes :
  - Annulation en cas d'échec : indique si la pile doit être annulée en cas d'échec de la création de la pile.
  - Délai d'expiration : nombre de minutes avant l'expiration du délai de création d'une pile.
  - Protection contre la terminaison : empêche la suppression accidentelle de la pile.

> **Note**
>
> AWS CloudFormation la protection contre la résiliation est différente du concept de protection contre la suppression d'Amazon DocumentDB. Pour plus d'informations, consultez Protection contre la résiliation et la suppression.

Choisissez Next (Suivant) pour continuer.

6. Révision <stack-name>: passez en revue votre modèle de stack, les détails et les options de configuration. Vous pouvez également ouvrir un lien de création rapide au bas de la page pour créer des piles avec les mêmes configurations de base que celle-ci.

   - Choisissez Créer pour créer la pile.
   - Vous pouvez également choisir Créer un jeu de modifications. Un jeu de modifications est un aperçu de la façon dont cette pile sera configurée avant de créer la pile. Cela vous permet d'examiner différentes configurations avant d'exécuter le jeu de modifications.

Accès au cluster Amazon DocumentDB

Une fois la AWS CloudFormation pile terminée, vous pouvez utiliser une instance Amazon EC2 pour vous connecter à votre cluster Amazon DocumentDB. Pour plus d'informations sur la connexion à une instance Amazon EC2 via SSH, consultez Connect to Your Linux Instance dans le guide de l'utilisateur Amazon EC2.

Une fois connecté, consultez les sections suivantes, qui contiennent des informations sur l'utilisation d'Amazon DocumentDB.

- Étape 4 : Installation du shell Mongo
- Suppression d'un cluster Amazon DocumentDB

Protection contre la résiliation et la suppression

L'une des meilleures pratiques d'Amazon DocumentDB consiste à activer la protection contre la suppression et la protection contre la résiliation. CloudFormation la protection contre la résiliation est une fonctionnalité nettement différente de la fonction de protection contre la suppression d'Amazon DocumentDB.

- Protection contre la résiliation : vous pouvez empêcher la suppression accidentelle d'une pile en activant la protection contre la résiliation pour votre CloudFormation pile. Si un utilisateur tente de supprimer une pile pour laquelle la protection contre la résiliation est activée, la suppression échoue et la pile demeure inchangée. La protection contre la résiliation est désactivée par défaut lorsque vous créez une pile à l'aide de CloudFormation. Vous pouvez activer la protection contre la résiliation sur une pile lorsque vous la créez. Pour plus d'informations, consultez la section Configuration des options de AWS CloudFormation pile.
- Protection contre la suppression — Amazon DocumentDB permet également d'activer la protection contre la suppression pour un cluster. Si un utilisateur tente de supprimer un cluster Amazon DocumentDB sur lequel la protection contre la suppression est activée, la suppression échoue et le cluster reste inchangé. La protection contre la suppression, lorsqu'elle est activée, protège contre les suppressions accidentelles d'Amazon AWS Management Console DocumentDB AWS CLI, et. CloudFormation Pour plus d'informations sur l'activation et la désactivation de la protection contre la suppression pour un cluster Amazon DocumentDB, consultez. Deletion protection (Protection contre la suppression)

Compatibilité avec MongoDB

Amazon DocumentDB prend en charge la compatibilité avec MongoDB, notamment MongoDB 4.0 et MongoDB 5.0. La compatibilité avec MongoDB signifie qu'une grande majorité des applications, pilotes et outils que vous utilisez déjà aujourd'hui avec vos bases de données MongoDB peuvent être utilisés avec Amazon DocumentDB avec peu ou pas de modifications. Cette section décrit tout ce que vous devez savoir sur la compatibilité d'Amazon DocumentDB avec MongoDB, notamment les nouvelles fonctionnalités, la mise en route, les chemins de migration et les différences fonctionnelles.

Rubriques

Compatibilité avec MongoDB 5.0

Rubriques

Nouveautés d'Amazon DocumentDB 5.0

Amazon DocumentDB 5.0 introduit de nouvelles fonctionnalités et capacités, notamment les limites de stockage et le chiffrement des champs côté client. Le résumé ci-dessous présente certaines des principales fonctionnalités introduites dans Amazon DocumentDB 5.0. Pour consulter la liste complète des nouvelles fonctionnalités, consultez leNotes de mise à jour.

- Limite de stockage augmentée à 128 TiB pour tous les clusters Amazon DocumentDB basés sur des instances et les clusters élastiques basés sur des partitions.
- Présentation du moteur Amazon DocumentDB 5.0 (version 3.0.775)
  - Support pour les pilotes d'API MongoDB 5.0

- Support du chiffrement au niveau du champ (FLE) côté client. Vous pouvez désormais chiffrer les champs côté client avant d'écrire les données dans le cluster Amazon DocumentDB. Pour plus d'informations, voir Chiffrement au niveau des champs côté client
- Nouveaux opérateurs d'agrégation :`$dateAdd`, `$dateSubtract`
- Supports pour les index avec `$elemMatch` opérateur. Par conséquent, les requêtes correspondantes `$elemMatch` entraîneront des analyses d'index.

Amazon DocumentDB ne prend pas en charge toutes les fonctionnalités de MongoDB 5.0. Lorsque nous avons créé Amazon DocumentDB 5.0, nous avons travaillé à rebours à partir des fonctionnalités et des fonctionnalités que nos clients nous ont le plus demandées. Nous continuerons à ajouter des fonctionnalités MongoDB 5.0 supplémentaires en fonction de ce que les clients nous demandent de créer. Pour consulter la dernière liste des API prises en charge, consultezAPI MongoDB, opérations et types de données pris en charge.

Commencez avec Amazon DocumentDB 5.0

Pour démarrer avec Amazon DocumentDB 5.0, consultez le guide de démarrage. Vous pouvez créer un nouveau cluster Amazon DocumentDB 5.0 à l'aide du AWS Management Console ou du AWS SDK, AWS CLI ou. AWS CloudFormation Lorsque vous vous connectez à Amazon DocumentDB, vous devez utiliser un pilote ou un utilitaire MongoDB compatible avec MongoDB 5.0 ou version ultérieure.

> **Note**
>
> Lorsque vous utilisez le AWS SDK, ou AWS CLI AWS CloudFormation, la version du moteur sera par défaut 5.0.0. Vous devez spécifier explicitement le paramètre `engineVersion = 4.0.0` pour créer un nouveau cluster Amazon DocumentDB 4.0 ou `engineVersion = 3.6.0` pour créer un nouveau cluster Amazon DocumentDB 3.6. Pour un cluster Amazon DocumentDB donné, vous pouvez déterminer la version du cluster à l'aide du AWS CLI to call `describe-db-clusters` ou utiliser la console de gestion Amazon DocumentDB pour afficher le numéro de version du moteur d'un cluster en particulier.

Amazon DocumentDB 5.0 prend en charge les processeurs Amazon EC2 Graviton2, `r6g` tels que les types d'instances pour vos clusters, `t4.medium` et est disponible dans toutes les régions prises en charge. Pour plus d'informations sur la tarification, consultez la section Tarification d'Amazon DocumentDB (compatible avec MongoDB).

Mise à niveau ou migration vers Amazon DocumentDB 4.0

Vous pouvez migrer de MongoDB 3.6 AWS DMSou MongoDB 4.0 vers Amazon DocumentDB 5.0 à l'aide des utilitaires tels que,, et. `mongodump` `mongorestore` `mongoimport` `mongoexport` Pour obtenir des instructions sur la façon de migrer, consultezMise à niveau de votre cluster Amazon DocumentDB à l'aide de AWS Database Migration Service.

Différences fonctionnelles

Différences fonctionnelles entre Amazon DocumentDB 4.0 et 5.0

Avec la sortie d'Amazon DocumentDB 5.0, il existe des différences fonctionnelles entre Amazon DocumentDB 3.6 et Amazon DocumentDB 4.0 :

- Le rôle intégré de sauvegarde est désormais compatible`serverStatus`. Action : les développeurs et les applications dotés d'un rôle de sauvegarde peuvent collecter des statistiques sur l'état du cluster Amazon DocumentDB.
- Le `SecondaryDelaySecs` champ est remplacé `slaveDelay` dans la `replSetGetConfig` sortie.
- La hello commande remplace `isMaster` - hello renvoie un document qui décrit le rôle d'un cluster Amazon DocumentDB.
- Amazon DocumentDB 5.0 prend désormais en charge les scans d'index avec l'`$elemMatch`opérateur au premier niveau d'imbrication. Les analyses d'index sont prises en charge lorsque le filtre réservé aux requêtes possède un niveau de `$elemMatch` filtre, mais elles ne sont pas prises en charge si une `$elemMatch` requête imbriquée est incluse.

  Par exemple, dans Amazon DocumentDB 5.0, si vous incluez l'`$elemMatch`opérateur dans le niveau imbriqué, il ne renverra pas de valeur comme dans Amazon DocumentDB 4.0 :

```
db.foo.insert(
[
    {a: {b: 5}},
    {a: {b: [5]}},
    {a: {b: [3, 7]}},
    {a: [{b: 5}]},
    {a: [{b: 3}, {b: 7}]},
    {a: [{b: [5]}]},
    {a: [{b: [3, 7]}]},
    {a: [[{b: 5}]]},
```

```
    {a: [[{b: 3}, {b: 7}]]},
    {a: [[{b: [5]}]]},
    {a: [[{b: [3, 7]}]]}
]);

// DocumentDB 5.0
> db.foo.find({a: {$elemMatch: {b: {$elemMatch: {$lt: 6, $gt: 4}}}}}, {_id: 0})
{ "a" : [ { "b" : [ 5 ] } ] }

// DocumentDB 4.0
> db.foo.find({a: {$elemMatch: {b: {$elemMatch: {$lt: 6, $gt: 4}}}}}, {_id: 0})
{ "a" : [ { "b" : [ 5 ] } ] }
{ "a" : [ [ { "b" : [ 5 ] } ] ] }
```

- La projection « $ » dans Amazon DocumentDB 4.0 renvoie tous les documents avec tous les champs. Avec Amazon DocumentDB 5.0, la find commande avec une projection « $ » renvoie les documents qui correspondent au paramètre de requête contenant uniquement le champ correspondant à la projection « $ ».
- Dans Amazon DocumentDB 5.0, find les commandes contenant les paramètres `$regex` et les paramètres de `$options` requête renvoient le message d'erreur suivant : « Impossible de définir les options dans les deux options `$regex` et `$options` ».
- Avec Amazon DocumentDB 5.0, renvoie `$indexOfCP` désormais « -1 » lorsque :
  - la sous-chaîne est introuvable dans l'expression de chaîne, ou
  - le début est un nombre supérieur à la fin, ou
  - start est un nombre supérieur à la longueur en octets de la chaîne.
- Dans Amazon DocumentDB 4.0, `$indexOfCP` renvoie « 0 » lorsque la position de départ est un nombre supérieur à la fin ou à la longueur en octets de la chaîne.
- Avec Amazon DocumentDB 5.0, les opérations de projection renvoient`_id fields`, par exemple`{"_id.nestedField" : 1}`, des documents qui incluent uniquement le champ projeté. Alors que dans Amazon DocumentDB 4.0, les commandes de projection de champs imbriqués ne filtrent aucun document.

Compatibilité avec MongoDB 4.0

Rubriques

- Mise à niveau ou migration vers Amazon DocumentDB 4.0
- Différences fonctionnelles

Fonctionnalités d'Amazon DocumentDB 4.0

Amazon DocumentDB 4.0 a introduit de nombreuses nouvelles fonctionnalités et capacités, notamment des transactions ACID et des améliorations apportées aux flux de modifications. Le résumé ci-dessous présente certaines des principales fonctionnalités introduites dans Amazon DocumentDB 4.0. Pour consulter la liste complète des fonctionnalités, consultez leNotes de mise à jour.

- Transactions ACID : Amazon DocumentDB permet désormais d'effectuer des transactions sur plusieurs documents, relevés, collections et bases de données. Les transactions simplifient le développement d'applications en vous permettant d'effectuer des opérations atomiques, cohérentes, isolées et durables (ACID) sur un ou plusieurs documents d'un cluster Amazon DocumentDB. Pour plus d'informations, consultez Transactions.
- Flux de modifications : vous pouvez désormais ouvrir un flux de modifications au niveau du cluster (`client.watch()`ou`mongo.watch()`) et de la base de données (`db.watch()`), vous pouvez spécifier un curseur `startAtOperationTime` pour ouvrir un flux de modifications, et enfin vous pouvez étendre la période de conservation de votre flux de modifications à 7 jours (24 heures auparavant). Pour plus d'informations, consultez Utilisation de Change Streams avec Amazon DocumentDB.
- AWS Database Migration Service(AWS DMS) : Vous pouvez désormais l'utiliser AWS DMS pour migrer vos charges de travail MongoDB 4.0 vers Amazon DocumentDB. AWS DMS prend désormais en charge une source MongoDB 4.0, une cible Amazon DocumentDB 4.0 et une source Amazon DocumentDB 3.6 pour effectuer des mises à niveau entre Amazon DocumentDB 3.6 et 4.0. Pour plus d'informations, consultez la documentation AWS DMS.
- Performances et indexation : vous pouvez désormais utiliser un index avec`$lookup`, rechercher des requêtes avec une projection contenant un champ ou un champ et le `_id` champ pouvant être diffusé directement à partir de l'index et sans avoir à lire dans la collection (requête couverte), la possibilité de le faire `hint()` avec`findAndModify`, des optimisations des performances et des améliorations visant à réduire la taille globale de l'index. `$addToSet` Pour plus d'informations, consultez Notes de mise à jour.
- Opérateurs : Amazon DocumentDB 4.0 prend désormais en charge un certain nombre de nouveaux opérateurs d'agrégation :`$ifNull`,,`$replaceRoot`,`$setIsSubset`,`$setInstersection`,`$setUnion`.

`$setEquals` Vous pouvez voir toutes les API, opérations et types de données MongoDB que nous prenons en charge sur. API MongoDB, opérations et types de données pris en charge

- Contrôle d'accès basé sur les rôles (RBAC) : avec les deux `ListDatabase` commandes `ListCollection` and, vous pouvez désormais éventuellement utiliser les `authorizedDatabases` paramètres `authorizedCollections` et pour permettre aux utilisateurs de répertorier les collections et les bases de données auxquelles ils sont autorisés à accéder sans avoir besoin des `listDatabase` rôles `listCollections` et, respectivement. Vous avez également la possibilité de tuer vos propres curseurs sans avoir besoin du `KillCursor` rôle.

Amazon DocumentDB ne prend pas en charge toutes les fonctionnalités de MongoDB 4.0. Lorsque nous avons créé Amazon DocumentDB 4.0, nous avons travaillé à rebours à partir des fonctionnalités et des fonctionnalités que nos clients nous ont le plus demandées. Nous continuerons à ajouter des fonctionnalités MongoDB 4.0 supplémentaires en fonction de ce que les clients nous demandent de créer. Par exemple, Amazon DocumentDB 4.0 ne prend actuellement pas en charge les opérateurs de conversion de type ou les opérateurs de chaîne introduits dans MongoDB 4.0. Pour consulter la dernière liste des API prises en charge, consultezAPI MongoDB, opérations et types de données pris en charge.

Commencez avec Amazon DocumentDB 4.0

Pour démarrer avec Amazon DocumentDB 4.0, consultez le guide de démarrage. Vous pouvez créer un nouveau cluster Amazon DocumentDB 4.0 à l'aide du AWS Management Console ou du AWS SDK, AWS CLI ou. AWS CloudFormation Lorsque vous vous connectez à Amazon DocumentDB, vous devez utiliser un pilote ou un utilitaire MongoDB compatible avec MongoDB 4.0 ou version ultérieure.

> **Note**
>
> Lorsque vous utilisez le AWS SDK, ou AWS CLI AWS CloudFormation, la version du moteur sera par défaut 5.0.0. Vous devez spécifier explicitement le paramètre `engineVersion = 4.0.0` pour créer un nouveau cluster Amazon DocumentDB 4.0 ou `engineVersion = 3.6.0` pour créer un nouveau cluster Amazon DocumentDB 3.6. Pour un cluster Amazon DocumentDB donné, vous pouvez déterminer la version du cluster à l'aide du AWS CLI to call `describe-db-clusters` ou utiliser la console de gestion Amazon DocumentDB pour afficher le numéro de version du moteur d'un cluster en particulier.

Amazon DocumentDB 4.0 prend en charge les types d'`t4g.medium`instance `r5` `r6gt3.medium`,, et pour vos clusters et est disponible dans toutes les régions prises en charge. L'utilisation d'Amazon DocumentDB 4.0 est gratuite. Pour plus d'informations sur la tarification, consultez la section Tarification d'Amazon DocumentDB (compatible avec MongoDB).

Mise à niveau ou migration vers Amazon DocumentDB 4.0

Vous pouvez migrer de MongoDB 3.6 AWS DMSou MongoDB 4.0 vers Amazon DocumentDB 4.0 à l'aide des utilitaires tels que,, et. `mongodump` `mongorestore` `mongoimport` `mongoexport` De même, vous pouvez utiliser les mêmes outils pour effectuer une mise à niveau d'Amazon DocumentDB 3.6 vers Amazon DocumentDB 4.0. Pour obtenir des instructions sur la façon de migrer, consultezMise à niveau de votre cluster Amazon DocumentDB à l'aide de AWS Database Migration Service.

Différences fonctionnelles

Différences fonctionnelles entre Amazon DocumentDB 3.6 et 4.0

Avec la sortie d'Amazon DocumentDB 4.0, il existe des différences fonctionnelles entre Amazon DocumentDB 3.6 et Amazon DocumentDB 4.0 :

- Projection pour les documents imbriqués : Amazon DocumentDB 3.6 prend en compte le premier champ d'un document imbriqué lors de l'application d'une projection. Cependant, Amazon DocumentDB 4.0 analysera les sous-documents et appliquera également la projection à chaque sous-document. Par exemple : si la projection l'est`"a.b.c": 1`, le comportement dans les deux versions est identique. Toutefois, si c'est le cas, `{a:{b:{c:1}}}` Amazon DocumentDB 3.6 appliquera la projection uniquement à « a » et non à « b » ou « c' ».
- Comportement pour**`minKey`**, **`maxKey`** : dans Amazon DocumentDB 4.0, le comportement pour ne rien `{x:{$gt:MaxKey}}` renvoyer et pour tout `{x:{$lt:MaxKey}}` renvoyer.
- Différences entre les documents : la comparaison de valeurs numériques de différents types (double, int, long) dans des sous-documents (par exemple, b in`{"_id" :1, "a" :{"b":1}}`) fournit désormais un résultat cohérent pour tous les types de données numériques et pour chaque niveau d'un document.

Différences fonctionnelles entre Amazon DocumentDB 4.0 et MongoDB 4.0

Vous trouverez ci-dessous les différences fonctionnelles entre Amazon DocumentDB 4.0 et MongoDB 4.0.

- Recherche avec une clé vide dans le chemin : lorsqu'une collection contient un document avec une clé vide dans le tableau (par exemple`{"x" : [ { "" : 10 }, { "b" : 20 } ]}`), et lorsque la clé utilisée dans la requête se termine par une chaîne vide (par exemple`x.`), Amazon DocumentDB renvoie ce document car il parcourt tous les documents du tableau alors que MongoDB ne renvoie pas ce document.
- **`$setOnInsert`**ainsi que **`$`** dans le chemin : l'opérateur de champ ne `$setOnInsert` fonctionnera pas en combinaison avec le chemin $ dans Amazon DocumentDB, qui est également compatible avec MongoDB 4.0.

Transactions

Amazon DocumentDB (compatible avec MongoDB) prend désormais en charge la compatibilité avec MongoDB 4.0, y compris les transactions. Vous pouvez effectuer des transactions sur plusieurs documents, relevés, collections et bases de données. Les transactions simplifient le développement d'applications en vous permettant d'effectuer des opérations atomiques, cohérentes, isolées et durables (ACID) sur un ou plusieurs documents au sein d'un cluster Amazon DocumentDB. Les cas d'utilisation courants des transactions incluent le traitement financier, l'exécution et la gestion des commandes et la création de jeux multijoueurs.

Il n'y a aucun coût supplémentaire pour les transactions. Vous ne payez que pour les iOS en lecture et en écriture que vous consommez dans le cadre des transactions.

Rubriques

Prérequis

Pour utiliser la fonction de transactions, vous devez répondre aux exigences suivantes :

- Vous devez utiliser le moteur Amazon DocumentDB 4.0.
- Vous devez utiliser un pilote compatible avec MongoDB 4.0 ou supérieur.

Bonnes pratiques

Voici quelques bonnes pratiques pour tirer le meilleur parti des transactions avec Amazon DocumentDB.

- Validez ou annulez toujours la transaction une fois qu'elle est terminée. Laisser une transaction dans un état incomplet monopolise les ressources de la base de données et peut entraîner des conflits d'écriture.
- Il est recommandé de limiter les transactions au plus petit nombre de commandes nécessaires. Si vous avez des transactions comportant plusieurs relevés qui peuvent être divisés en plusieurs transactions plus petites, il est conseillé de le faire afin de réduire la probabilité d'un délai d'attente. Essayez toujours de créer des transactions courtes, et non des lectures de longue durée.

Limites

- Amazon DocumentDB ne prend pas en charge les curseurs dans une transaction.
- Amazon DocumentDB ne peut pas créer de nouvelles collections dans une transaction et ne peut pas interroger/mettre à jour des collections inexistantes.
- Les verrous d'écriture au niveau du document sont soumis à un délai d'expiration d'une minute, qui n'est pas configurable par l'utilisateur.
- Les commandes d'écriture réessayable, de validation réessayable et d'abandon réessayable ne sont pas prises en charge dans Amazon DocumentDB. Exception : Si vous utilisez mongo shell, n'incluez la`retryWrites=false` commande dans aucune chaîne de code. Par défaut, les écritures réessayables sont désactivées. L'inclusion`retryWrites=false` peut entraîner l'échec des commandes de lecture normales.
- Chaque instance Amazon DocumentDB possède une limite supérieure quant au nombre de transactions simultanées ouvertes simultanément sur l'instance. Pour les limites, veuillez consulterLimites d’instance.
- Pour une transaction donnée, la taille du journal des transactions doit être inférieure à 32 Mo.
- Amazon DocumentDB prend en charge`count()` les transactions au sein d'une transaction, mais tous les pilotes ne prennent pas en charge cette fonctionnalité. Une alternative consiste à utiliser l'`countDocuments()`API, qui traduit la requête de comptage en une requête d'agrégation côté client.

- Les transactions ont une limite d'exécution d'une minute et les sessions ont un délai d'expiration de 30 minutes. Si une transaction expire, elle sera abandonnée et toutes les commandes ultérieures émises au cours de la session pour la transaction existante produiront l'erreur suivante :

```
WriteCommandError({
"ok" : 0,
"operationTime" : Timestamp(1603491424, 627726),
"code" : 251,
"errmsg" : "Given transaction number 0 does not match any in-progress transactions."
})
```

Surveillance et diagnostic

Avec la prise en charge des transactions dans Amazon DocumentDB 4.0, des CloudWatch indicateurs supplémentaires ont été ajoutés pour vous aider à surveiller vos transactions.

Nouveaux CloudWatch indicateurs

- `DatabaseTransactions`: le nombre de transactions ouvertes effectuées sur une période d'une minute.
- `DatabaseTransactionsAborted`: le nombre de transactions abandonnées effectuées par période d'une minute.
- `DatabaseTransactionsMax`: le nombre maximum de transactions ouvertes sur une période d'une minute.
- `TransactionsAborted`: le nombre de transactions abandonnées sur une instance sur une période d'une minute.
- `TransactionsCommitted`: le nombre de transactions effectuées sur une instance sur une période d'une minute.
- `TransactionsOpen`: le nombre de transactions ouvertes sur une instance prise par période d'une minute.
- `TransactionsOpenMax`: le nombre maximum de transactions ouvertes sur une instance par période d'une minute.
- `TransactionsStarted`: le nombre de transactions démarrées sur une instance sur une période d'une minute.

> **Note**
>
> Pour en savoir plus sur les CloudWatch statistiques relatives à Amazon DocumentDB, rendez-vous surSurveillance d'Amazon DocumentDB avec CloudWatch.

De plus, de nouveaux champs ont été ajoutés aux deux`currentOplsidtransactionThreadId`, et un nouvel état pour «`idle transaction` » et`serverStatus` les transactions :`currentActive``currentInactive``currentOpen`,`totalAborted`,`totalCommitted`, et`totalStarted`.

Niveau d'isolement des transactions

Lorsque vous démarrez une transaction, il est possible de spécifier à la fois le`readConcern` et`writeConcern` comme indiqué dans l'exemple ci-dessous :

```
mySession.startTransaction({readConcern: {level: 'snapshot'}, writeConcern:
{w: 'majority'}});
```

En effet`readConcern`, Amazon DocumentDB prend en charge l'isolation des instantanés par défaut. Si une`readConcern` valeur locale, disponible ou majoritaire est spécifiée, Amazon DocumentDB passera au`readConcern` niveau instantané. Amazon DocumentDB ne prend pas en charge le linéarisable`readConcern` et la spécification d'un tel problème de lecture entraînera une erreur.

En effet`writeConcern`, Amazon DocumentDB prend en charge la majorité par défaut et un quorum d'écriture est atteint lorsque quatre copies des données sont conservées sur trois AZ. Si une valeur inférieure`writeConcern` est spécifiée, Amazon DocumentDB passe`writeConcern` à la majorité. En outre, toutes les écritures Amazon DocumentDB sont journalisées et la journalisation ne peut pas être désactivée.

Cas d'utilisation

Dans cette section, nous passerons en revue deux cas d'utilisation des transactions : plusieurs relevés et plusieurs collectes.

Transactions comportant plusieurs états

Les transactions Amazon DocumentDB comportent plusieurs déclarations, ce qui signifie que vous pouvez écrire une transaction qui couvre plusieurs instructions avec un commit ou un rollback

explicite. Vous pouvez regrouper`insert`,`updatedelete`, et des`findAndModify` actions en une seule opération atomique.

Un cas d'utilisation courant pour les transactions comportant plusieurs états est celui d'une transaction débit/crédit. Par exemple : vous devez de l'argent à un ami pour des vêtements. Ainsi, vous devez débiter (retirer) 500$ de votre compte et créditer 500$ (dépôt) sur le compte de votre ami. Pour effectuer cette opération, vous effectuez à la fois les opérations de dette et de crédit dans le cadre d'une seule transaction afin de garantir l'atomicité. Cela permet d'éviter les scénarios où 500$ sont débités de votre compte, mais pas crédités sur le compte de votre ami. Voici à quoi ressemblerait ce cas d'utilisation :

```
// *** Transfer $500 from Alice to Bob inside a transaction: Success Scenario***
// Setup bank account for Alice and Bob. Each have $1000 in their account

var databaseName = "bank";
var collectionName = "account";
var amountToTransfer = 500;

var session = db.getMongo().startSession({causalConsistency: false});
var bankDB = session.getDatabase(databaseName);
var accountColl = bankDB[collectionName];
accountColl.drop();

accountColl.insert({name: "Alice", balance: 1000});
accountColl.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountColl.find({"name": "Alice"}).next().balance;
var newAliceBalance = aliceBalance - amountToTransfer;
accountColl.update({"name": "Alice"},{"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountColl.find({"name": "Alice"}).next().balance;

// add $500 to Bob's account
var bobBalance = accountColl.find({"name": "Bob"}).next().balance;
var newBobBalance = bobBalance + amountToTransfer;
accountColl.update({"name": "Bob"},{"$set": {"balance": newBobBalance}});
var findBobBalance = accountColl.find({"name": "Bob"}).next().balance;

session.commitTransaction();
```

```
accountColl.find();

// *** Transfer $500 from Alice to Bob inside a transaction: Failure Scenario***

// Setup bank account for Alice and Bob. Each have $1000 in their account
var databaseName = "bank";
var collectionName = "account";
var amountToTransfer = 500;

var session = db.getMongo().startSession({causalConsistency: false});
var bankDB = session.getDatabase(databaseName);
var accountColl = bankDB[collectionName];
accountColl.drop();

accountColl.insert({name: "Alice", balance: 1000});
accountColl.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountColl.find({"name": "Alice"}).next().balance;
var newAliceBalance = aliceBalance - amountToTransfer;
accountColl.update({"name": "Alice"},{"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountColl.find({"name": "Alice"}).next().balance;

session.abortTransaction();
```

Transactions à collectes multiples

Nos transactions sont également multi-collections, ce qui signifie qu'elles peuvent être utilisées pour effectuer plusieurs opérations au sein d'une même transaction et sur plusieurs collections. Cela fournit une vue cohérente des données et préserve l'intégrité de vos données. Lorsque vous validez les commandes en une seule<>, les transactions sont all-or-nothing des exécutions, c'est-à-dire qu'elles réussiront toutes ou échoueront toutes.

Voici un exemple de transactions à encaissements multiples, utilisant le même scénario et les mêmes données que l'exemple pour les transactions à relevés multiples.

```
// *** Transfer $500 from Alice to Bob inside a transaction: Success Scenario***
```

```
// Setup bank account for Alice and Bob. Each have $1000 in their account
var amountToTransfer = 500;
var collectionName = "account";

var session = db.getMongo().startSession({causalConsistency: false});
var accountCollInBankA = session.getDatabase("bankA")[collectionName];
var accountCollInBankB = session.getDatabase("bankB")[collectionName];

accountCollInBankA.drop();
accountCollInBankB.drop();

accountCollInBankA.insert({name: "Alice", balance: 1000});
accountCollInBankB.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountCollInBankA.find({"name": "Alice"}).next().balance;
var newAliceBalance = aliceBalance - amountToTransfer;
accountCollInBankA.update({"name": "Alice"},{"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountCollInBankA.find({"name": "Alice"}).next().balance;

// add $500 to Bob's account
var bobBalance = accountCollInBankB.find({"name": "Bob"}).next().balance;
var newBobBalance = bobBalance + amountToTransfer;
accountCollInBankB.update({"name": "Bob"},{"$set": {"balance": newBobBalance}});
var findBobBalance = accountCollInBankB.find({"name": "Bob"}).next().balance;

session.commitTransaction();

accountCollInBankA.find(); // Alice holds $500 in bankA
accountCollInBankB.find(); // Bob holds $1500 in bankB

// *** Transfer $500 from Alice to Bob inside a transaction: Failure Scenario***

// Setup bank account for Alice and Bob. Each have $1000 in their account
var collectionName = "account";
var amountToTransfer = 500;

var session = db.getMongo().startSession({causalConsistency: false});
var accountCollInBankA = session.getDatabase("bankA")[collectionName];
var accountCollInBankB = session.getDatabase("bankB")[collectionName];
```

```
accountCollInBankA.drop();
accountCollInBankB.drop();

accountCollInBankA.insert({name: "Alice", balance: 1000});
accountCollInBankB.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountCollInBankA.find({"name": "Alice"}).next().balance;
var newAliceBalance = aliceBalance - amountToTransfer;
accountCollInBankA.update({"name": "Alice"},{"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountCollInBankA.find({"name": "Alice"}).next().balance;

// add $500 to Bob's account
var bobBalance = accountCollInBankB.find({"name": "Bob"}).next().balance;
var newBobBalance = bobBalance + amountToTransfer;
accountCollInBankB.update({"name": "Bob"},{"$set": {"balance": newBobBalance}});
var findBobBalance = accountCollInBankB.find({"name": "Bob"}).next().balance;

session.abortTransaction();

accountCollInBankA.find(); // Alice holds $1000 in bankA
accountCollInBankB.find(); // Bob holds $1000 in bankB
```

Exemples d'API de transaction pour l'API de rappel

L'API de rappel n'est disponible que pour les pilotes 4.2+.

Javascript

Le code suivant montre comment utiliser l'API de transaction Amazon DocumentDB avec Javascript.

```
// *** Transfer $500 from Alice to Bob inside a transaction: Success ***
// Setup bank account for Alice and Bob. Each have $1000 in their account
var databaseName = "bank";
var collectionName = "account";
var amountToTransfer = 500;

var session = db.getMongo().startSession({causalConsistency: false});
var bankDB = session.getDatabase(databaseName);
```

```
var accountColl = bankDB[collectionName];
accountColl.drop();

accountColl.insert({name: "Alice", balance: 1000});
accountColl.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountColl.find({"name": "Alice"}).next().balance;
assert(aliceBalance >= amountToTransfer);
var newAliceBalance = aliceBalance - amountToTransfer;
accountColl.update({"name": "Alice"},{"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountColl.find({"name": "Alice"}).next().balance;
assert.eq(newAliceBalance, findAliceBalance);

// add $500 to Bob's account
var bobBalance = accountColl.find({"name": "Bob"}).next().balance;
var newBobBalance = bobBalance + amountToTransfer;
accountColl.update({"name": "Bob"},{"$set": {"balance": newBobBalance}});
var findBobBalance = accountColl.find({"name": "Bob"}).next().balance;
assert.eq(newBobBalance, findBobBalance);

session.commitTransaction();

accountColl.find();
```

Node.js

Le code suivant montre comment utiliser l'API de transaction Amazon DocumentDB avec Node.js.

```
// Node.js callback API:

const bankDB = await mongoclient.db("bank");
var accountColl = await bankDB.createCollection("account");
var amountToTransfer = 500;

const session = mongoclient.startSession({causalConsistency: false});
await accountColl.drop();

await accountColl.insertOne({name: "Alice", balance: 1000}, { session });
await accountColl.insertOne({name: "Bob", balance: 1000}, { session });

const transactionOptions = {
```

```
    readConcern: { level: 'snapshot' },
    writeConcern: { w: 'majority' }
    };

// deduct $500 from Alice's account
var aliceBalance = await accountColl.findOne({name: "Alice"}, {session});
assert(aliceBalance.balance >= amountToTransfer);
var newAliceBalance = aliceBalance - amountToTransfer;
session.startTransaction(transactionOptions);
await accountColl.updateOne({name: "Alice"}, {$set: {balance: newAliceBalance}},
 {session });
await session.commitTransaction();
aliceBalance = await accountColl.findOne({name: "Alice"}, {session});
assert(newAliceBalance == aliceBalance.balance);

// add $500 to Bob's account
var bobBalance = await accountColl.findOne({name: "Bob"}, {session});
var newBobBalance = bobBalance.balance + amountToTransfer;
session.startTransaction(transactionOptions);
await accountColl.updateOne({name: "Bob"}, {$set: {balance: newBobBalance}},
 {session });
await session.commitTransaction();
bobBalance = await accountColl.findOne({name: "Bob"}, {session});
assert(newBobBalance == bobBalance.balance);
```

C#

Le code suivant montre comment utiliser l'API de transaction Amazon DocumentDB avec C#.

```
// C# Callback API

var dbName = "bank";
var collName = "account";
var amountToTransfer = 500;

using (var session = client.StartSession(new ClientSessionOptions{CausalConsistency
 = false}))
{
    var bankDB = client.GetDatabase(dbName);
    var accountColl = bankDB.GetCollection<BsonDocument>(collName);
    bankDB.DropCollection(collName);
    accountColl.InsertOne(session, new BsonDocument { {"name", "Alice"}, {"balance",
 1000 } });
```

```
   accountColl.InsertOne(session, new BsonDocument { {"name", "Bob"}, {"balance",
1000 } });

   // start transaction
   var transactionOptions = new TransactionOptions(
           readConcern: ReadConcern.Snapshot,
           writeConcern: WriteConcern.WMajority);
   var result = session.WithTransaction(
       (sess, cancellationtoken) =>
       {
           // deduct $500 from Alice's account
           var aliceBalance = accountColl.Find(sess,
Builders<BsonDocument>.Filter.Eq("name",
"Alice")).FirstOrDefault().GetValue("balance");
           Debug.Assert(aliceBalance >= amountToTransfer);
           var newAliceBalance = aliceBalance.AsInt32 - amountToTransfer;
           accountColl.UpdateOne(sess, Builders<BsonDocument>.Filter.Eq("name",
"Alice"),
                                 Builders<BsonDocument>.Update.Set("balance",
newAliceBalance));
           aliceBalance = accountColl.Find(sess,
Builders<BsonDocument>.Filter.Eq("name",
"Alice")).FirstOrDefault().GetValue("balance");
           Debug.Assert(aliceBalance == newAliceBalance);

           // add $500 from Bob's account
           var bobBalance = accountColl.Find(sess,
Builders<BsonDocument>.Filter.Eq("name",
"Bob")).FirstOrDefault().GetValue("balance");
           var newBobBalance = bobBalance.AsInt32 + amountToTransfer;
           accountColl.UpdateOne(sess, Builders<BsonDocument>.Filter.Eq("name",
"Bob"),
                                 Builders<BsonDocument>.Update.Set("balance",
newBobBalance));
           bobBalance = accountColl.Find(sess,
Builders<BsonDocument>.Filter.Eq("name",
"Bob")).FirstOrDefault().GetValue("balance");
           Debug.Assert(bobBalance == newBobBalance);

           return "Transaction committed";
       }, transactionOptions);
  // check values outside of transaction
   var aliceNewBalance = accountColl.Find(Builders<BsonDocument>.Filter.Eq("name",
"Alice")).FirstOrDefault().GetValue("balance");
```

```
    var bobNewBalance = accountColl.Find(Builders<BsonDocument>.Filter.Eq("name",
 "Bob")).FirstOrDefault().GetValue("balance");
    Debug.Assert(aliceNewBalance ==  500);
    Debug.Assert(bobNewBalance ==  1500);
}
```

Ruby

Le code suivant montre comment utiliser l'API de transaction Amazon DocumentDB avec Ruby.

```
// Ruby Callback API

dbName = "bank"
collName = "account"
amountToTransfer = 500

session = client.start_session(:causal_consistency=> false)
bankDB = Mongo::Database.new(client, dbName)
accountColl = bankDB[collName]
accountColl.drop()

accountColl.insert_one({"name"=>"Alice", "balance"=>1000})
accountColl.insert_one({"name"=>"Bob", "balance"=>1000})

    # start transaction
    session.with_transaction(read_concern: {level: :snapshot}, write_concern:
 {w: :majority}) do
        # deduct $500 from Alice's account
        aliceBalance = accountColl.find({"name"=>"Alice"}, :session=>
 session).first['balance']
        assert aliceBalance >= amountToTransfer
        newAliceBalance = aliceBalance - amountToTransfer
        accountColl.update_one({"name"=>"Alice"}, { "$set" =>
 {"balance"=>newAliceBalance} }, :session=> session)
        aliceBalance = accountColl.find({"name"=>>"Alice"}, :session=>
 session).first['balance']
        assert_equal(newAliceBalance, aliceBalance)

        # add $500 from Bob's account
        bobBalance = accountColl.find({"name"=>"Bob"}, :session=>
 session).first['balance']
        newBobBalance = bobBalance + amountToTransfer
        accountColl.update_one({"name"=>"Bob"}, { "$set" =>
 {"balance"=>newBobBalance} }, :session=> session)
```

```
        bobBalance = accountColl.find({"name"=>"Bob"}, :session=>
 session).first['balance']
        assert_equal(newBobBalance, bobBalance)
    end

   # check results outside of transaction
    aliceBalance = accountColl.find({"name"=>"Alice"}).first['balance']
    bobBalance = accountColl.find({"name"=>"Bob"}).first['balance']
    assert_equal(aliceBalance, 500)
    assert_equal(bobBalance, 1500)

session.end_session
```

Go

Le code suivant montre comment utiliser l'API de transaction Amazon DocumentDB avec Go.

```
// Go - Callback API
type Account struct {
    Name string
    Balance  int
}

ctx := context.TODO()

dbName := "bank"
collName := "account"
amountToTransfer := 500

session, err := client.StartSession(options.Session().SetCausalConsistency(false))
assert.NilError(t, err)
defer session.EndSession(ctx)

bankDB := client.Database(dbName)
accountColl := bankDB.Collection(collName)
accountColl.Drop(ctx)

_, err = accountColl.InsertOne(ctx, bson.M{"name" : "Alice", "balance":1000})
_, err = accountColl.InsertOne(ctx, bson.M{"name" : "Bob", "balance":1000})

transactionOptions := options.Transaction().SetReadConcern(readconcern.Snapshot()).

 SetWriteConcern(writeconcern.New(writeconcern.WMajority()))
```

```
_, err = session.WithTransaction(ctx, func(sessionCtx mongo.SessionContext)
 (interface{}, error) {
    var result Account
    // deduct $500 from Alice's account
    err = accountColl.FindOne(sessionCtx, bson.M{"name": "Alice"}).Decode(&result)
    aliceBalance := result.Balance
    newAliceBalance := aliceBalance - amountToTransfer
    _, err = accountColl.UpdateOne(sessionCtx, bson.M{"name": "Alice"},
 bson.M{"$set": bson.M{"balance": newAliceBalance}})
    err = accountColl.FindOne(sessionCtx, bson.M{"name": "Alice"}).Decode(&result)
    aliceBalance = result.Balance
    assert.Equal(t, aliceBalance, newAliceBalance)

    // add $500 to Bob's account
    err = accountColl.FindOne(sessionCtx, bson.M{"name": "Bob"}).Decode(&result)
    bobBalance := result.Balance
    newBobBalance := bobBalance + amountToTransfer
    _, err = accountColl.UpdateOne(sessionCtx, bson.M{"name": "Bob"}, bson.M{"$set":
 bson.M{"balance": newBobBalance}})
    err = accountColl.FindOne(sessionCtx, bson.M{"name": "Bob"}).Decode(&result)
    bobBalance = result.Balance
    assert.Equal(t, bobBalance, newBobBalance)

    if err != nil {
        return nil, err
    }
    return "transaction committed", err
}, transactionOptions)

// check results outside of transaction
var result Account
err = accountColl.FindOne(ctx, bson.M{"name": "Alice"}).Decode(&result)
aliceNewBalance := result.Balance
err = accountColl.FindOne(ctx, bson.M{"name": "Bob"}).Decode(&result)
bobNewBalance := result.Balance
assert.Equal(t, aliceNewBalance, 500)
assert.Equal(t, bobNewBalance, 1500)
// Go - Core API
type Account struct {
    Name string
    Balance  int
}
```

```
func transferMoneyWithRetry(sessionContext mongo.SessionContext, accountColl
 *mongo.Collection, t *testing.T) error {
    amountToTransfer := 500

    transactionOptions :=
 options.Transaction().SetReadConcern(readconcern.Snapshot()).

 SetWriteConcern(writeconcern.New(writeconcern.WMajority()))
    if err := sessionContext.StartTransaction(transactionOptions); err != nil {
        panic(err)
    }

    var result Account
    // deduct $500 from Alice's account
    err := accountColl.FindOne(sessionContext, bson.M{"name":
 "Alice"}).Decode(&result)
    aliceBalance := result.Balance
    newAliceBalance := aliceBalance - amountToTransfer
    _, err = accountColl.UpdateOne(sessionContext, bson.M{"name": "Alice"},
 bson.M{"$set": bson.M{"balance": newAliceBalance}})
    if err != nil {
        sessionContext.AbortTransaction(sessionContext)
    }
    err = accountColl.FindOne(sessionContext, bson.M{"name":
 "Alice"}).Decode(&result)
    aliceBalance = result.Balance
    assert.Equal(t, aliceBalance, newAliceBalance)

    // add $500 to Bob's account
    err = accountColl.FindOne(sessionContext, bson.M{"name": "Bob"}).Decode(&result)
    bobBalance := result.Balance
    newBobBalance := bobBalance + amountToTransfer
    _, err = accountColl.UpdateOne(sessionContext, bson.M{"name": "Bob"},
 bson.M{"$set": bson.M{"balance": newBobBalance}})
    if err != nil {
        sessionContext.AbortTransaction(sessionContext)
    }
    err = accountColl.FindOne(sessionContext, bson.M{"name": "Bob"}).Decode(&result)
    bobBalance = result.Balance
    assert.Equal(t, bobBalance, newBobBalance)

    err = sessionContext.CommitTransaction(sessionContext)
    return err
}
```

```
func doTransactionWithRetry(t *testing.T) {
    ctx := context.TODO()

    dbName := "bank"
    collName := "account"
    bankDB := client.Database(dbName)
    accountColl := bankDB.Collection(collName)

    client.UseSessionWithOptions(ctx, options.Session().SetCausalConsistency(false),
 func(sessionContext mongo.SessionContext) error {
        accountColl.Drop(ctx)
        accountColl.InsertOne(sessionContext, bson.M{"name" : "Alice",
 "balance":1000})
        accountColl.InsertOne(sessionContext, bson.M{"name" : "Bob",
 "balance":1000})
        for {
            err := transferMoneyWithRetry(sessionContext, accountColl, t)
            if err == nil {
                println("transaction committed")
                return nil
            }
            if mongoErr := err.(mongo.CommandError);
 mongoErr.HasErrorLabel("TransientTransactionError") {
                continue
            }
            println("transaction failed")
            return err
        }
    })

    // check results outside of transaction
    var result Account
    accountColl.FindOne(ctx, bson.M{"name": "Alice"}).Decode(&esult)
    aliceBalance := result.Balance
    assert.Equal(t, aliceBalance, 500)
    accountColl.FindOne(ctx, bson.M{"name": "Bob"}).Decode(&result)
    bobBalance := result.Balance
    assert.Equal(t, bobBalance, 1500)
}
```

Java

Le code suivant montre comment utiliser l'API de transaction Amazon DocumentDB avec Java.

```
// Java (sync) - Callback API
MongoDatabase bankDB = mongoClient.getDatabase("bank");
MongoCollection accountColl = bankDB.getCollection("account");
accountColl.drop();
int amountToTransfer = 500;

// add sample data
accountColl.insertOne(new Document("name", "Alice").append("balance", 1000));
accountColl.insertOne(new Document("name", "Bob").append("balance", 1000));

TransactionOptions txnOptions = TransactionOptions.builder()
        .readConcern(ReadConcern.SNAPSHOT)
        .writeConcern(WriteConcern.MAJORITY)
        .build();
ClientSessionOptions sessionOptions =
 ClientSessionOptions.builder().causallyConsistent(false).build();
try ( ClientSession clientSession = mongoClient.startSession(sessionOptions) ) {
    clientSession.withTransaction(new TransactionBody<Void>() {
        @Override
        public Void execute() {
            // deduct $500 from Alice's account
            List<Document> documentList = new ArrayList<>();
            accountColl.find(clientSession, new Document("name",
 "Alice")).into(documentList);
            int aliceBalance = (int) documentList.get(0).get("balance");
            int newAliceBalance = aliceBalance - amountToTransfer;

            accountColl.updateOne(clientSession, new Document("name", "Alice"), new
 Document("$set", new Document("balance", newAliceBalance)));

            // check Alice's new balance
            documentList = new ArrayList<>();
            accountColl.find(clientSession, new Document("name",
 "Alice")).into(documentList);
            int updatedBalance = (int) documentList.get(0).get("balance");
            Assert.assertEquals(updatedBalance, newAliceBalance);

            // add $500 to Bob's account
            documentList = new ArrayList<>();
            accountColl.find(clientSession, new Document("name",
 "Bob")).into(documentList);
            int bobBalance = (int) documentList.get(0).get("balance");
            int newBobBalance = bobBalance + amountToTransfer;
```

```
            accountColl.updateOne(clientSession, new Document("name", "Bob"), new
 Document("$set", new Document("balance", newBobBalance)));

            // check Bob's new balance
            documentList = new ArrayList<>();
            accountColl.find(clientSession, new Document("name",
 "Bob")).into(documentList);
            updatedBalance = (int) documentList.get(0).get("balance");
            Assert.assertEquals(updatedBalance, newBobBalance);

            return null;
        }
    }, txnOptions);
}
```

C

Le code suivant montre comment utiliser l'API de transaction Amazon DocumentDB avec C.

```
// Sample Code for C with Callback

#include <bson.h>
#include <mongoc.h>
#include <stdio.h>
#include <string.h>
#include <assert.h>

typedef struct {
    int64_t balance;
    bson_t *account;
    bson_t *opts;
    mongoc_collection_t *collection;
} ctx_t;

bool callback_session (mongoc_client_session_t *session, void *ctx, bson_t **reply,
 bson_error_t *error)
{
    bool r = true;
    ctx_t *data = (ctx_t *) ctx;
    bson_t local_reply;
    bson_t *selector = data->account;
    bson_t *update = BCON_NEW ("$set", "{", "balance", BCON_INT64 (data->balance),
 "}");
```

```
    mongoc_collection_update_one (data->collection, selector, update, data->opts,
 &local_reply, error);

    *reply = bson_copy (&local_reply);
    bson_destroy (&local_reply);
    bson_destroy (update);
    return r;
}

void test_callback_money_transfer(mongoc_client_t* client, mongoc_collection_t*
 collection, int amount_to_transfer){

    bson_t reply;
    bool r = true;
    const bson_t *doc;
    bson_iter_t iter;
    ctx_t alice_ctx;
    ctx_t bob_ctx;
    bson_error_t error;

    // find query
    bson_t *alice_query = bson_new ();
    BSON_APPEND_UTF8(alice_query, "name", "Alice");

    bson_t *bob_query = bson_new ();
    BSON_APPEND_UTF8(bob_query, "name", "Bob");

    // create session
    // set causal consistency to false
    mongoc_session_opt_t *session_opts = mongoc_session_opts_new ();
    mongoc_session_opts_set_causal_consistency (session_opts, false);
    // start the session
    mongoc_client_session_t *client_session = mongoc_client_start_session (client,
 session_opts, &error);

    // add session to options
    bson_t *opts = bson_new();
    mongoc_client_session_append (client_session, opts, &error);

    // deduct 500 from Alice
    // find account balance of Alice
    mongoc_cursor_t *cursor = mongoc_collection_find_with_opts (collection,
 alice_query, NULL, NULL);
```

```
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    int64_t alice_balance = (bson_iter_value (&iter))->value.v_int64;
    assert(alice_balance >= amount_to_transfer);
    int64_t new_alice_balance = alice_balance - amount_to_transfer;

    // set variables which will be used by callback function
    alice_ctx.collection = collection;
    alice_ctx.opts = opts;
    alice_ctx.balance = new_alice_balance;
    alice_ctx.account = alice_query;

    // callback
    r = mongoc_client_session_with_transaction (client_session, &callback_session,
NULL, &alice_ctx, &reply, &error);
    assert(r);

    // find account balance of Alice after transaction
    cursor = mongoc_collection_find_with_opts (collection, alice_query, NULL, NULL);
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    alice_balance = (bson_iter_value (&iter))->value.v_int64;
    assert(alice_balance == new_alice_balance);
    assert(alice_balance == 500);

        // add 500 to bob's balance
    // find account balance of Bob
    cursor = mongoc_collection_find_with_opts (collection, bob_query, NULL, NULL);
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    int64_t bob_balance = (bson_iter_value (&iter))->value.v_int64;
    int64_t new_bob_balance = bob_balance + amount_to_transfer;

    bob_ctx.collection = collection;
    bob_ctx.opts = opts;
    bob_ctx.balance = new_bob_balance;
    bob_ctx.account = bob_query;

    // set read & write concern
    mongoc_read_concern_t *read_concern = mongoc_read_concern_new ();
    mongoc_write_concern_t *write_concern = mongoc_write_concern_new ();
```

```
    mongoc_transaction_opt_t *txn_opts = mongoc_transaction_opts_new ();

    mongoc_write_concern_set_w(write_concern, MONGOC_WRITE_CONCERN_W_MAJORITY);
    mongoc_read_concern_set_level(read_concern, MONGOC_READ_CONCERN_LEVEL_SNAPSHOT);
    mongoc_transaction_opts_set_write_concern (txn_opts, write_concern);
    mongoc_transaction_opts_set_read_concern (txn_opts, read_concern);

    // callback
    r = mongoc_client_session_with_transaction (client_session, &callback_session,
 txn_opts, &bob_ctx, &reply, &error);
    assert(r);

 // find account balance of Bob after transaction
    cursor = mongoc_collection_find_with_opts (collection, bob_query, NULL, NULL);
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    bob_balance = (bson_iter_value (&iter))->value.v_int64;
    assert(bob_balance == new_bob_balance);
    assert(bob_balance == 1500);

    // cleanup
    bson_destroy(alice_query);
    bson_destroy(bob_query);
    mongoc_client_session_destroy(client_session);
    bson_destroy(opts);
    mongoc_transaction_opts_destroy(txn_opts);
    mongoc_read_concern_destroy(read_concern);
    mongoc_write_concern_destroy(write_concern);
    mongoc_cursor_destroy(cursor);
    bson_destroy(doc);
}
int main(int argc, char* argv[]) {
    mongoc_init ();
    mongoc_client_t* client = mongoc_client_new (<connection uri>);
    bson_error_t error;

    // connect to bank db
    mongoc_database_t *database = mongoc_client_get_database (client, "bank");
    // access account collection
    mongoc_collection_t* collection = mongoc_client_get_collection(client, "bank",
 "account");
    // set amount to transfer
    int64_t amount_to_transfer = 500;
```

```
    // delete the collection if already existing
    mongoc_collection_drop(collection, &error);

    // open Alice account
    bson_t *alice_account = bson_new ();
    BSON_APPEND_UTF8(alice_account, "name", "Alice");
    BSON_APPEND_INT64(alice_account, "balance", 1000);

    // open Bob account
    bson_t *bob_account = bson_new ();
    BSON_APPEND_UTF8(bob_account, "name", "Bob");
    BSON_APPEND_INT64(bob_account, "balance", 1000);

    bool r = true;

    r = mongoc_collection_insert_one(collection, alice_account, NULL, NULL, &error);
    if (!r) {printf("Error encountered:%s", error.message);}
    r = mongoc_collection_insert_one(collection, bob_account, NULL, NULL, &error);
    if (!r) {printf("Error encountered:%s", error.message);}

    test_callback_money_transfer(client, collection, amount_to_transfer);

}
```

Python

Le code suivant montre comment utiliser l'API de transaction Amazon DocumentDB avec Python.

```
// Sample Python code with callback api

import pymongo

def callback(session, balance, query):
    collection.update_one(query, {'$set': {"balance": balance}}, session=session)

client = pymongo.MongoClient(<connection uri>)
rc_snapshot = pymongo.read_concern.ReadConcern('snapshot')
wc_majority = pymongo.write_concern.WriteConcern('majority')

# To start, drop and create an account collection and insert balances for both Alice
 and Bob
collection = client.get_database("bank").get_collection("account")
collection.drop()
collection.insert_one({"_id": 1, "name": "Alice", "balance": 1000})
```

```
collection.insert_one({"_id": 2, "name": "Bob", "balance": 1000})

amount_to_transfer = 500

# deduct 500 from Alice's account
alice_balance = collection.find_one({"name": "Alice"}).get("balance")
assert alice_balance >= amount_to_transfer
new_alice_balance = alice_balance - amount_to_transfer

with client.start_session({'causalConsistency':False}) as session:
    session.with_transaction(lambda s: callback(s, new_alice_balance, {"name":
 "Alice"}), read_concern=rc_snapshot, write_concern=wc_majority)

updated_alice_balance = collection.find_one({"name": "Alice"}).get("balance")
assert updated_alice_balance == new_alice_balance

# add 500 to Bob's account
bob_balance = collection.find_one({"name": "Bob"}).get("balance")
assert bob_balance >= amount_to_transfer
new_bob_balance = bob_balance + amount_to_transfer

with client.start_session({'causalConsistency':False}) as session:
    session.with_transaction(lambda s: callback(s, new_bob_balance, {"name":
 "Bob"}), read_concern=rc_snapshot, write_concern=wc_majority)

updated_bob_balance = collection.find_one({"name": "Bob"}).get("balance")
assert updated_bob_balance == new_bob_balance
Sample Python code with Core api
import pymongo

client = pymongo.MongoClient(<connection_string>)
rc_snapshot = pymongo.read_concern.ReadConcern('snapshot')
wc_majority = pymongo.write_concern.WriteConcern('majority')

# To start, drop and create an account collection and insert balances for both Alice
 and Bob
collection = client.get_database("bank").get_collection("account")
collection.drop()
collection.insert_one({"_id": 1, "name": "Alice", "balance": 1000})
collection.insert_one({"_id": 2, "name": "Bob", "balance": 1000})

amount_to_transfer = 500

# deduct 500 from Alice's account
```

```
alice_balance = collection.find_one({"name": "Alice"}).get("balance")
assert alice_balance >= amount_to_transfer
new_alice_balance = alice_balance - amount_to_transfer

with client.start_session({'causalConsistency':False}) as session:
    session.start_transaction(read_concern=rc_snapshot, write_concern=wc_majority)
    collection.update_one({"name": "Alice"}, {'$set': {"balance":
 new_alice_balance}}, session=session)
    session.commit_transaction()

updated_alice_balance = collection.find_one({"name": "Alice"}).get("balance")
assert updated_alice_balance == new_alice_balance

# add 500 to Bob's account
bob_balance = collection.find_one({"name": "Bob"}).get("balance")
assert bob_balance >= amount_to_transfer
new_bob_balance = bob_balance + amount_to_transfer

with client.start_session({'causalConsistency':False}) as session:
    session.start_transaction(read_concern=rc_snapshot, write_concern=wc_majority)
    collection.update_one({"name": "Bob"}, {'$set': {"balance": new_bob_balance}},
 session=session)
    session.commit_transaction()

updated_bob_balance = collection.find_one({"name": "Bob"}).get("balance")
assert updated_bob_balance == new_bob_balance
```

Exemples d'API de transaction pour l'API principale

Javascript

Le code suivant montre comment utiliser l'API de transaction Amazon DocumentDB avec Javascript.

```
// *** Transfer $500 from Alice to Bob inside a transaction: Success ***
// Setup bank account for Alice and Bob. Each have $1000 in their account
var databaseName = "bank";
var collectionName = "account";
var amountToTransfer = 500;

var session = db.getMongo().startSession({causalConsistency: false});
var bankDB = session.getDatabase(databaseName);
```

```
var accountColl = bankDB[collectionName];
accountColl.drop();

accountColl.insert({name: "Alice", balance: 1000});
accountColl.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountColl.find({"name": "Alice"}).next().balance;
assert(aliceBalance >= amountToTransfer);
var newAliceBalance = aliceBalance - amountToTransfer;
accountColl.update({"name": "Alice"},{"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountColl.find({"name": "Alice"}).next().balance;
assert.eq(newAliceBalance, findAliceBalance);

// add $500 to Bob's account
var bobBalance = accountColl.find({"name": "Bob"}).next().balance;
var newBobBalance = bobBalance + amountToTransfer;
accountColl.update({"name": "Bob"},{"$set": {"balance": newBobBalance}});
var findBobBalance = accountColl.find({"name": "Bob"}).next().balance;
assert.eq(newBobBalance, findBobBalance);

session.commitTransaction();

accountColl.find();
```

C#

Le code suivant montre comment utiliser l'API de transaction Amazon DocumentDB avec C#.

```
// C# Core API

public void TransferMoneyWithRetry(IMongoCollection<bSondocument> accountColl,
 IClientSessionHandle session)
{
    var amountToTransfer = 500;

    // start transaction
   var transactionOptions = new TransactionOptions(
               readConcern: ReadConcern.Snapshot,
               writeConcern: WriteConcern.WMajority);
    session.StartTransaction(transactionOptions);
   try
```

```
    {
        // deduct $500 from Alice's account
        var aliceBalance = accountColl.Find(session,
 Builders<bSondocument>.Filter.Eq("name",
 "Alice")).FirstOrDefault().GetValue("balance");
        Debug.Assert(aliceBalance >= amountToTransfer);
        var newAliceBalance = aliceBalance.AsInt32 - amountToTransfer;
        accountColl.UpdateOne(session, Builders<bSondocument>.Filter.Eq("name",
 "Alice"),
                                Builders<bSondocument>.Update.Set("balance",
 newAliceBalance));
        aliceBalance = accountColl.Find(session,
 Builders<bSondocument>.Filter.Eq("name",
 "Alice")).FirstOrDefault().GetValue("balance");
        Debug.Assert(aliceBalance == newAliceBalance);

        // add $500 from Bob's account
        var bobBalance = accountColl.Find(session,
 Builders<bSondocument>.Filter.Eq("name",
 "Bob")).FirstOrDefault().GetValue("balance");
        var newBobBalance = bobBalance.AsInt32 + amountToTransfer;
        accountColl.UpdateOne(session, Builders<bSondocument>.Filter.Eq("name",
 "Bob"),
                                Builders<bSondocument>.Update.Set("balance",
 newBobBalance));
        bobBalance = accountColl.Find(session,
 Builders<bSondocument>.Filter.Eq("name",
 "Bob")).FirstOrDefault().GetValue("balance");
        Debug.Assert(bobBalance == newBobBalance);

    }
    catch (Exception e)
    {
        session.AbortTransaction();
        throw;
    }

    session.CommitTransaction();
 }

}
public void DoTransactionWithRetry(MongoClient client)
{
    var dbName = "bank";
```

```
   var collName = "account";
   using (var session = client.StartSession(new
ClientSessionOptions{CausalConsistency = false}))
   {
       try
       {
           var bankDB = client.GetDatabase(dbName);
           var accountColl = bankDB.GetCollection<bSondocument>(collName);
           bankDB.DropCollection(collName);
           accountColl.InsertOne(session, new BsonDocument { {"name", "Alice"},
{"balance", 1000 } });
           accountColl.InsertOne(session, new BsonDocument { {"name", "Bob"},
{"balance", 1000 } });

           while(true) {
               try
               {
                       TransferMoneyWithRetry(accountColl, session);
                       break;
               }
               catch (MongoException e)
               {
                   if(e.HasErrorLabel("TransientTransactionError"))
                   {
                       continue;
                   }
                   else
                   {
                       throw;
                   }
               }
           }

           // check values outside of transaction
           var aliceNewBalance =
accountColl.Find(Builders<bSondocument>.Filter.Eq("name",
"Alice")).FirstOrDefault().GetValue("balance");
           var bobNewBalance =
accountColl.Find(Builders<bSondocument>.Filter.Eq("name",
"Bob")).FirstOrDefault().GetValue("balance");
           Debug.Assert(aliceNewBalance ==  500);
           Debug.Assert(bobNewBalance ==  1500);
       }
       catch (Exception e)
```

```
        {
            Console.WriteLine("Error running transaction: " + e.Message);
        }
    }
}
```

Ruby

Le code suivant montre comment utiliser l'API de transaction Amazon DocumentDB avec Ruby.

```
# Ruby Core API

def transfer_money_w_retry(session, accountColl)
    amountToTransfer = 500

    session.start_transaction(read_concern: {level: :snapshot}, write_concern:
 {w: :majority})
    # deduct $500 from Alice's account
    aliceBalance = accountColl.find({"name"=>"Alice"}, :session=>
 session).first['balance']
    assert aliceBalance >= amountToTransfer
    newAliceBalance = aliceBalance - amountToTransfer
    accountColl.update_one({"name"=>"Alice"}, { "$set" =>
 {"balance"=>newAliceBalance} }, :session=> session)
    aliceBalance = accountColl.find({"name"=>"Alice"}, :session=>
 session).first['balance']
    assert_equal(newAliceBalance, aliceBalance)

    # add $500 to Bob's account
    bobBalance = accountColl.find({"name"=>"Bob"}, :session=>
 session).first['balance']
    newBobBalance = bobBalance + amountToTransfer
    accountColl.update_one({"name"=>"Bob"}, { "$set" =>
 {"balance"=>newBobBalance} }, :session=> session)
    bobBalance = accountColl.find({"name"=>"Bob"}, :session=>
 session).first['balance']
    assert_equal(newBobBalance, bobBalance)

    session.commit_transaction

end

def do_txn_w_retry(client)
     dbName = "bank"
```

```
    collName = "account"

    session = client.start_session(:causal_consistency=> false)
    bankDB = Mongo::Database.new(client, dbName)
    accountColl = bankDB[collName]
    accountColl.drop()

    accountColl.insert_one({"name"=>"Alice", "balance"=>1000})
    accountColl.insert_one({"name"=>"Bob", "balance"=>1000})

    begin
        transferMoneyWithRetry(session, accountColl)
        puts "transaction committed"
    rescue Mongo::Error => e
        if e.label?('TransientTransactionError')
            retry
        else
            puts "transaction failed"
            raise
        end
    end

    # check results outside of transaction
    aliceBalance = accountColl.find({"name"=>"Alice"}).first['balance']
    bobBalance = accountColl.find({"name"=>"Bob"}).first['balance']
    assert_equal(aliceBalance, 500)
    assert_equal(bobBalance, 1500)

end
```

Java

Le code suivant montre comment utiliser l'API de transaction Amazon DocumentDB avec Java.

```
// Java (sync) - Core API

public void transferMoneyWithRetry() {
   // connect to server
    MongoClientURI mongoURI = new MongoClientURI(uri);
    MongoClient mongoClient = new MongoClient(mongoURI);

    MongoDatabase bankDB = mongoClient.getDatabase("bank");
    MongoCollection accountColl = bankDB.getCollection("account");
    accountColl.drop();
```

```
   // insert some sample data
    accountColl.insertOne(new Document("name", "Alice").append("balance", 1000));
    accountColl.insertOne(new Document("name", "Bob").append("balance", 1000));

    while (true) {
        try {
            doTransferMoneyWithRetry(accountColl, mongoClient);
            break;
        } catch (MongoException e) {
            if (e.hasErrorLabel(MongoException.TRANSIENT_TRANSACTION_ERROR_LABEL)) {
                continue;
            } else {
                throw e;
            }
        }
    }
}

public void doTransferMoneyWithRetry(MongoCollection accountColl, MongoClient
 mongoClient) {
    int amountToTransfer = 500;

   TransactionOptions txnOptions = TransactionOptions.builder()
      .readConcern(ReadConcern.SNAPSHOT)
      .writeConcern(WriteConcern.MAJORITY)
      .build();
    ClientSessionOptions sessionOptions =
 ClientSessionOptions.builder().causallyConsistent(false).build();
    try ( ClientSession clientSession = mongoClient.startSession(sessionOptions) ) {
        clientSession.startTransaction(txnOptions);

        // deduct $500 from Alice's account
        List<Document> documentList = new ArrayList<>();
        accountColl.find(clientSession, new Document("name",
 "Alice")).into(documentList);
        int aliceBalance = (int) documentList.get(0).get("balance");
        Assert.assertTrue(aliceBalance >= amountToTransfer);
        int newAliceBalance = aliceBalance - amountToTransfer;
        accountColl.updateOne(clientSession, new Document("name", "Alice"), new
 Document("$set", new Document("balance", newAliceBalance)));

        // check Alice's new balance
        documentList = new ArrayList<>();
```

```
        accountColl.find(clientSession, new Document("name",
 "Alice")).into(documentList);
        int updatedBalance = (int) documentList.get(0).get("balance");
        Assert.assertEquals(updatedBalance, newAliceBalance);

        // add $500 to Bob's account
        documentList = new ArrayList<>();
        accountColl.find(clientSession, new Document("name",
 "Bob")).into(documentList);
        int bobBalance = (int) documentList.get(0).get("balance");
        int newBobBalance = bobBalance + amountToTransfer;
        accountColl.updateOne(clientSession, new Document("name", "Bob"), new
 Document("$set", new Document("balance", newBobBalance)));

        // check Bob's new balance
        documentList = new ArrayList<>();
        accountColl.find(clientSession, new Document("name",
 "Bob")).into(documentList);
        updatedBalance = (int) documentList.get(0).get("balance");
        Assert.assertEquals(updatedBalance, newBobBalance);

        // commit transaction
        clientSession.commitTransaction();
    }
}
// Java (async) -- Core API
public void transferMoneyWithRetry() {
    // connect to the server
    MongoClient mongoClient = MongoClients.create(uri);

    MongoDatabase bankDB = mongoClient.getDatabase("bank");
    MongoCollection accountColl = bankDB.getCollection("account");
    SubscriberLatchWrapper<Void> dropCallback = new SubscriberLatchWrapper<>();
    mongoClient.getDatabase("bank").drop().subscribe(dropCallback);
    dropCallback.await();

    // insert some sample data
    SubscriberLatchWrapper<InsertOneResult> insertionCallback = new
 SubscriberLatchWrapper<>();
    accountColl.insertOne(new Document("name", "Alice").append("balance",
 1000)).subscribe(insertionCallback);
    insertionCallback.await();

    insertionCallback = new SubscriberLatchWrapper<>();
```

```
    accountColl.insertOne(new Document("name", "Bob").append("balance",
 1000)).subscribe(insertionCallback);;
    insertionCallback.await();

    while (true) {
        try {
            doTransferMoneyWithRetry(accountColl, mongoClient);
            break;
        } catch (MongoException e) {
            if (e.hasErrorLabel(MongoException.TRANSIENT_TRANSACTION_ERROR_LABEL)) {
                continue;
            } else {
                throw e;
            }
        }
    }
}

public void doTransferMoneyWithRetry(MongoCollection accountColl, MongoClient
 mongoClient) {
    int amountToTransfer = 500;

    // start the transaction
    TransactionOptions txnOptions = TransactionOptions.builder()
            .readConcern(ReadConcern.SNAPSHOT)
            .writeConcern(WriteConcern.MAJORITY)
            .build();
    ClientSessionOptions sessionOptions =
 ClientSessionOptions.builder().causallyConsistent(false).build();

    SubscriberLatchWrapper<ClientSession> sessionCallback = new
 SubscriberLatchWrapper<>();
    mongoClient.startSession(sessionOptions).subscribe(sessionCallback);
    ClientSession session = sessionCallback.get().get(0);
    session.startTransaction(txnOptions);

    // deduct $500 from Alice's account
    SubscriberLatchWrapper<Document> findCallback = new SubscriberLatchWrapper<>();
    accountColl.find(session, new Document("name",
 "Alice")).first().subscribe(findCallback);
    Document documentFound = findCallback.get().get(0);
    int aliceBalance = (int) documentFound.get("balance");
    int newAliceBalance = aliceBalance - amountToTransfer;
```

```
    SubscriberLatchWrapper<UpdateResult> updateCallback = new
 SubscriberLatchWrapper<>();
    accountColl.updateOne(session, new Document("name",
 "Alice"), new Document("$set", new Document("balance",
 newAliceBalance))).subscribe(updateCallback);
    updateCallback.await();

    // check Alice's new balance
    findCallback = new SubscriberLatchWrapper<>();
    accountColl.find(session, new Document("name",
 "Alice")).first().subscribe(findCallback);
    documentFound = findCallback.get().get(0);
    int updatedBalance = (int) documentFound.get("balance");
    Assert.assertEquals(updatedBalance, newAliceBalance);

    // add $500 to Bob's account
    findCallback = new SubscriberLatchWrapper<>();
    accountColl.find(session, new Document("name",
 "Bob")).first().subscribe(findCallback);
    documentFound = findCallback.get().get(0);
    int bobBalance = (int) documentFound.get("balance");
    int newBobBalance = bobBalance + amountToTransfer;

    updateCallback = new SubscriberLatchWrapper<>();
    accountColl.updateOne(session, new Document("name", "Bob"), new Document("$set",
 new Document("balance", newBobBalance))).subscribe(updateCallback);
    updateCallback.await();

    // check Bob's new balance
    findCallback = new SubscriberLatchWrapper<>();
    accountColl.find(session, new Document("name",
 "Bob")).first().subscribe(findCallback);
    documentFound = findCallback.get().get(0);
    updatedBalance = (int) documentFound.get("balance");
    Assert.assertEquals(updatedBalance, newBobBalance);

    // commit the transaction
    SubscriberLatchWrapper<Void> transactionCallback = new
 SubscriberLatchWrapper<>();
    session.commitTransaction().subscribe(transactionCallback);
    transactionCallback.await();
}

public class SubscriberLatchWrapper<T> implements Subscriber<T> {
```

```
    /**
     * A Subscriber that stores the publishers results and provides a latch so can
block on completion.
     *
     * @param <T> The publishers result type
     */
    private final List<T> received;
    private final List<RuntimeException> errors;
    private final CountDownLatch latch;
    private volatile Subscription subscription;
    private volatile boolean completed;

    /**
     * Construct an instance
     */
    public SubscriberLatchWrapper() {
        this.received = new ArrayList<>();
        this.errors = new ArrayList<>();
        this.latch = new CountDownLatch(1);
    }

    @Override
    public void onSubscribe(final Subscription s) {
        subscription = s;
        subscription.request(Integer.MAX_VALUE);
    }

    @Override
    public void onNext(final T t) {
        received.add(t);
    }

    @Override
    public void onError(final Throwable t) {
        if (t instanceof RuntimeException) {
            errors.add((RuntimeException) t);
        } else {
            errors.add(new RuntimeException("Unexpected exception", t));
        }
        onComplete();
    }

    @Override
```

```
    public void onComplete() {
        completed = true;
        subscription.cancel();
        latch.countDown();
    }

    /**
     * Get received elements
     *
     * @return the list of received elements
     */
    public List<T> getReceived() {
        return received;
    }

    /**
     * Get received elements.
     *
     * @return the list of receive elements
     */
    public List<T> get() {
        return await().getReceived();
    }

    /**
     * Await completion or error
     *
     * @return this
     */
    public SubscriberLatchWrapper<T> await() {
        subscription.request(Integer.MAX_VALUE);
        try {
            if (!latch.await(300, TimeUnit.SECONDS)) {
                throw new MongoTimeoutException("Publisher onComplete timed out for
300 seconds");
            }
        } catch (InterruptedException e) {
            throw new MongoInterruptedException("Interrupted waiting for
observeration", e);
        }
        if (!errors.isEmpty()) {
            throw errors.get(0);
        }
        return this;
```

```
    }

    public boolean getCompleted() {
        return this.completed;
    }

    public void close() {
        subscription.cancel();
        received.clear();
    }
}
```

C

Le code suivant montre comment utiliser l'API de transaction Amazon DocumentDB avec C.

```
// Sample C code with core session

bool core_session(mongoc_client_session_t *client_session, mongoc_collection_t*
 collection, bson_t *selector, int64_t balance){
    bool r = true;
    bson_error_t error;
    bson_t *opts = bson_new();
    bson_t *update = BCON_NEW ("$set", "{", "balance", BCON_INT64 (balance), "}");

    // set read & write concern
    mongoc_read_concern_t *read_concern = mongoc_read_concern_new ();
    mongoc_write_concern_t *write_concern = mongoc_write_concern_new ();
    mongoc_transaction_opt_t *txn_opts = mongoc_transaction_opts_new ();

    mongoc_write_concern_set_w(write_concern, MONGOC_WRITE_CONCERN_W_MAJORITY);
    mongoc_read_concern_set_level(read_concern, MONGOC_READ_CONCERN_LEVEL_SNAPSHOT);
    mongoc_transaction_opts_set_write_concern (txn_opts, write_concern);
    mongoc_transaction_opts_set_read_concern (txn_opts, read_concern);

    mongoc_client_session_start_transaction (client_session, txn_opts, &error);
    mongoc_client_session_append (client_session, opts, &error);

    r = mongoc_collection_update_one (collection, selector, update, opts, NULL,
 &error);

    mongoc_client_session_commit_transaction (client_session, NULL, &error);
    bson_destroy (opts);
```

```
    mongoc_transaction_opts_destroy(txn_opts);
    mongoc_read_concern_destroy(read_concern);
    mongoc_write_concern_destroy(write_concern);
    bson_destroy (update);
    return r;
}

void test_core_money_transfer(mongoc_client_t* client, mongoc_collection_t*
 collection, int amount_to_transfer){

    bson_t reply;
    bool r = true;
    const bson_t *doc;
    bson_iter_t iter;
    bson_error_t error;

    // find query
    bson_t *alice_query = bson_new ();
    BSON_APPEND_UTF8(alice_query, "name", "Alice");

    bson_t *bob_query = bson_new ();
    BSON_APPEND_UTF8(bob_query, "name", "Bob");

    // create session
    // set causal consistency to false
    mongoc_session_opt_t *session_opts = mongoc_session_opts_new ();
    mongoc_session_opts_set_causal_consistency (session_opts, false);
    // start the session
    mongoc_client_session_t *client_session = mongoc_client_start_session (client,
 session_opts, &error);

    // add session to options
    bson_t *opts = bson_new();
    mongoc_client_session_append (client_session, opts, &error);

    // deduct 500 from Alice
    // find account balance of Alice
    mongoc_cursor_t *cursor = mongoc_collection_find_with_opts (collection,
 alice_query, NULL, NULL);
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    int64_t alice_balance = (bson_iter_value (&iter))->value.v_int64;
    assert(alice_balance >= amount_to_transfer);
```

```
    int64_t new_alice_balance = alice_balance - amount_to_transfer;

    // core
    r = core_session (client_session, collection, alice_query, new_alice_balance);
    assert(r);

    // find account balance of Alice after transaction
    cursor = mongoc_collection_find_with_opts (collection, alice_query, NULL, NULL);
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    alice_balance = (bson_iter_value (&iter))->value.v_int64;
    assert(alice_balance == new_alice_balance);
    assert(alice_balance == 500);

    // add 500 to Bob's balance
    // find account balance of Bob
    cursor = mongoc_collection_find_with_opts (collection, bob_query, NULL, NULL);
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    int64_t bob_balance = (bson_iter_value (&iter))->value.v_int64;
    int64_t new_bob_balance = bob_balance + amount_to_transfer;

    //core
    r = core_session (client_session, collection, bob_query, new_bob_balance);
    assert(r);

    // find account balance of Bob after transaction
    cursor = mongoc_collection_find_with_opts (collection, bob_query, NULL, NULL);
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    bob_balance = (bson_iter_value (&iter))->value.v_int64;
    assert(bob_balance == new_bob_balance);
    assert(bob_balance == 1500);

    // cleanup
    bson_destroy(alice_query);
    bson_destroy(bob_query);
    mongoc_client_session_destroy(client_session);
    bson_destroy(opts);
    mongoc_cursor_destroy(cursor);
    bson_destroy(doc);
```

```
}

int main(int argc, char* argv[]) {
    mongoc_init ();
    mongoc_client_t* client = mongoc_client_new (<connection uri>);
    bson_error_t error;

    // connect to bank db
    mongoc_database_t *database = mongoc_client_get_database (client, "bank");
    // access account collection
    mongoc_collection_t* collection = mongoc_client_get_collection(client, "bank",
 "account");
    // set amount to transfer
    int64_t amount_to_transfer = 500;
    // delete the collection if already existing
    mongoc_collection_drop(collection, &error);

    // open Alice account
    bson_t *alice_account = bson_new ();
    BSON_APPEND_UTF8(alice_account, "name", "Alice");
    BSON_APPEND_INT64(alice_account, "balance", 1000);

    // open Bob account
    bson_t *bob_account = bson_new ();
    BSON_APPEND_UTF8(bob_account, "name", "Bob");
    BSON_APPEND_INT64(bob_account, "balance", 1000);

    bool r = true;

    r = mongoc_collection_insert_one(collection, alice_account, NULL, NULL, &error);
    if (!r) {printf("Error encountered:%s", error.message);}
    r = mongoc_collection_insert_one(collection, bob_account, NULL, NULL, &error);
    if (!r) {printf("Error encountered:%s", error.message);}

    test_core_money_transfer(client, collection, amount_to_transfer);

}
```

Scala

Le code suivant montre comment utiliser l'API de transaction Amazon DocumentDB avec Scala.

```
// Scala Core API
```

```
def transferMoneyWithRetry(sessionObservable: SingleObservable[ClientSession] ,
 database: MongoDatabase ): Unit = {
    val accountColl = database.getCollection("account")
    var amountToTransfer = 500

    var transactionObservable: Observable[ClientSession] =
 sessionObservable.map(clientSession => {
    clientSession.startTransaction()

    // deduct $500 from Alice's account
    var aliceBalance = accountColl.find(clientSession, Document("name" ->
 "Alice")).await().head.getInteger("balance")
    assert(aliceBalance >= amountToTransfer)
    var newAliceBalance = aliceBalance - amountToTransfer
    accountColl.updateOne(clientSession, Document("name" -> "Alice"),
 Document("$set" -> Document("balance" -> newAliceBalance))).await()
    aliceBalance = accountColl.find(clientSession, Document("name" ->
 "Alice")).await().head.getInteger("balance")
    assert(aliceBalance == newAliceBalance)

    // add $500 to Bob's account
    var bobBalance = accountColl.find(clientSession, Document("name" ->
 "Bob")).await().head.getInteger("balance")
    var newBobBalance = bobBalance + amountToTransfer
    accountColl.updateOne(clientSession, Document("name" -> "Bob"), Document("$set"
 -> Document("balance" -> newBobBalance))).await()
    bobBalance = accountColl.find(clientSession, Document("name" ->
 "Bob")).await().head.getInteger("balance")
    assert(bobBalance == newBobBalance)

    clientSession
    })

    transactionObservable.flatMap(clientSession =>
 clientSession.commitTransaction()).await()
}

def doTransactionWithRetry(): Unit = {
    val client: MongoClient = MongoClientWrapper.getMongoClient()
    val database: MongoDatabase = client.getDatabase("bank")
    val accountColl = database.getCollection("account")
    accountColl.drop().await()
```

```
    val sessionOptions =
 ClientSessionOptions.builder().causallyConsistent(false).build()
    var  sessionObservable: SingleObservable[ClientSession] =
 client.startSession(sessionOptions)
    accountColl.insertOne(Document("name" -> "Alice", "balance" -> 1000)).await()
    accountColl.insertOne(Document("name" -> "Bob", "balance" -> 1000)).await()

    var retry = true
    while (retry) {
        try {
        transferMoneyWithRetry(sessionObservable, database)
        println("transaction committed")
        retry = false
        }
        catch {
        case e: MongoException if
 e.hasErrorLabel(MongoException.TRANSIENT_TRANSACTION_ERROR_LABEL) => {
            println("retrying transaction")
        }
        case other: Throwable => {
            println("transaction failed")
            retry = false
            throw other

        }
        }
    }

    // check results outside of transaction
    assert(accountColl.find(Document("name" ->
 "Alice")).results().head.getInteger("balance") == 500)
    assert(accountColl.find(Document("name" ->
 "Bob")).results().head.getInteger("balance") == 1500)

    accountColl.drop().await()

}
```

Commandes prises en charge

Commande	Pris en charge
`abortTransaction`	Oui
`commitTransaction`	Oui
`endSessions`	Oui
`killSession`	Oui
`killAllSession`	Oui
`killAllSessionsByPattern`	Non
`refreshSessions`	Non
`startSession`	Oui

Fonctionnalités non prises en charge

Méthodes	Étapes ou commandes

`db.collection.aggregate()`	`$collStats`
	`$currentOp`
	`$indexStats`
	`$listSessions`
	`$out`

`db.collection.count()`
`db.collection.countDocuments()`
	`$where`
	`$near`
	`$nearSphere`

Méthodes	Étapes ou commandes
`db.collection.insert()`	`insert`n'est pas pris en charge s'il n'est pas exécuté sur une collection existante. Cette méthode est prise en charge si elle cible une collection préexistante.

Séances

Les sessions MongoDB sont un framework utilisé pour prendre en charge les écritures réessayables, la cohérence causale, les transactions et la gestion des opérations entre les bases de données. Lorsqu'une session est créée, un identifiant de session logique (lsid) est généré par le client et est utilisé pour étiqueter toutes les opérations au sein de cette session lors de l'envoi de commandes au serveur.

Amazon DocumentDB prend en charge l'utilisation de sessions pour activer les transactions, mais ne prend pas en charge la cohérence causale ni les écritures réessayables.

Lorsque vous utilisez des transactions dans Amazon DocumentDB, une transaction est initiée depuis une session à l'aide de l'`session.startTransaction()`API et une session prend en charge une seule transaction à la fois. De même, les transactions sont effectuées à l'aide des API commit (`session.commitTransaction()`) ou abort (`session.abortTransaction()`).

Cohérence causale

La cohérence causale garantit qu'au cours d'une seule session client, le client observera la read-after-write cohérence, les lectures/écritures monoatomiques et les écritures suivront les lectures et ces garanties s'appliquent à toutes les instances d'un cluster, et pas seulement à la principale. Amazon DocumentDB ne prend pas en charge la cohérence causale et la déclaration suivante provoquera une erreur.

```
var mySession = db.getMongo().startSession();
var mySessionObject = mySession.getDatabase('test').getCollection('account');

mySessionObject.updateOne({"_id": 2}, {"$inc": {"balance": 400}});
//Result:{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

```
mySessionObject.find()
//Error: error: {
//         "ok" : 0,
//         "code" : 303,
//         "errmsg" : "Feature not supported: 'causal consistency'",
//         "operationTime" : Timestamp(1603461817, 493214)
//}

mySession.endSession()
```

Vous pouvez désactiver la cohérence causale au sein d'une session. Veuillez noter que cela vous permettra d'utiliser le cadre de session, mais ne fournira aucune garantie de cohérence causale pour les lectures. Lorsque vous utilisez Amazon DocumentDB, les lectures à partir de la base principale seront read-after-write cohérentes et les lectures à partir des instances de réplication finiront par être cohérentes. Les transactions constituent le principal cas d'utilisation des sessions.

```
var mySession = db.getMongo().startSession({causalConsistency: false});
var mySessionObject = mySession.getDatabase('test').getCollection('account');

mySessionObject.updateOne({"_id": 2}, {"$inc": {"balance": 400}});
//Result:{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

mySessionObject.find()
//{ "_id" : 1, "name" : "Bob", "balance" : 100 }
//{ "_id" : 2, "name" : "Alice", "balance" : 1700 }
```

écritures réessayables

Les écritures réessayables sont une fonctionnalité grâce à laquelle le client tente de réessayer les opérations d'écriture, une seule fois, lorsque des erreurs réseau se produisent ou s'il ne parvient pas à trouver l'écriture principale. Dans Amazon DocumentDB, les écritures réessayables ne sont pas prises en charge et doivent être désactivées. Vous pouvez le désactiver à l'aide de la commande (`retryWrites=false`) dans la chaîne de connexion.

Exception : Si vous utilisez mongo shell, n'incluez la`retryWrites=false` commande dans aucune chaîne de code. Par défaut, les écritures réessayables sont désactivées. L'inclusion`retryWrites=false` peut entraîner l'échec des commandes de lecture normales.

Erreurs de transaction

Lorsque vous utilisez des transactions, certains scénarios peuvent générer une erreur indiquant qu'un numéro de transaction ne correspond à aucune transaction en cours.

L'erreur peut être générée dans au moins deux scénarios différents :

- After the one-minute transaction timeout.
- After an instance restart (due to patching, crash recovery, etc.), it is possible to receive this error even in cases where the transaction successfully committed. During an instance restart, the database can't tell the difference between a transaction that successfully completed versus a transaction that aborted. In other words, the transaction completion state is ambiguous.

La meilleure façon de gérer cette erreur est de rendre les mises à jour transactionnelles idempotentes, par exemple en utilisant le`$set` mutateur au lieu d'une opération d'incrémentation/décrémentation. Voir ci-dessous :

```
{ "ok" : 0,
"operationTime" : Timestamp(1603938167, 1),
"code" : 251,
"errmsg" : "Given transaction number 1 does not match any in-progress transactions."
}
```

Bonnes pratiques pour Amazon DocumentDB

Découvrez les meilleures pratiques pour travailler avec Amazon DocumentDB (compatible avec MongoDB). Cette section est mise à jour en continu à mesure que de nouvelles bonnes pratiques sont identifiées.

Rubriques

Directives opérationnelles de base

Voici les directives opérationnelles de base que tout le monde doit suivre lorsqu'il travaille avec Amazon DocumentDB. L'accord de niveau de service Amazon DocumentDB exige que vous suiviez ces directives.

- Déployez un cluster composé d'au moins deux instances Amazon DocumentDB dans deux zones de AWS disponibilité. Pour les charges de travail de production, nous recommandons de déployer un cluster composé d'au moins trois instances Amazon DocumentDB dans trois zones de disponibilité.
- Utilisez le service dans les limites de service préconisées. Pour plus d’informations, consultez Quotas et limites Amazon DocumentDB.
- Surveillez votre mémoire, votre processeur, vos connexions et votre utilisation du stockage. Pour vous aider à maintenir les performances et la disponibilité du système, configurez Amazon

CloudWatch pour qu'il vous avertisse lorsque les habitudes d'utilisation changent ou lorsque vous approchez de la capacité de votre déploiement.

- Augmentez la capacité de votre instance lorsque vous atteignez la limite de stockage. Vos instances doivent être provisionnées avec suffisamment de ressources de calcul (RAM, UC) pour répondre aux augmentations imprévues de la demande de vos applications.
- Définissez la période de conservation des sauvegardes en fonction de votre objectif de point de récupération.
- Testez le basculement pour votre cluster afin de connaître la durée du processus pour votre cas d'utilisation. Pour plus d'informations, consultez Basculement Amazon DocumentDB.
- Connectez-vous à votre cluster Amazon DocumentDB avec le point de terminaison du cluster (voirPoints de terminaison Amazon DocumentDB) et en mode Replica Set (voirConnexion à Amazon DocumentDB en tant qu'ensemble de réplicas) afin de minimiser l'impact d'un basculement sur votre application.
- Choisissez un mode de préférence de lecture de pilote qui optimise la disponibilité en lecture tout en répondant à vos exigences de cohérence en lecture de votre application. La préférence de `secondaryPreferred` lecture active les lectures de réplica et libère l'instance principale pour effectuer plus de travail. Pour plus d'informations, consultez Options de préférence de lecture.
- Concevez votre application pour qu'elle soit résiliente en cas d'erreurs réseau et de base de données. Utilisez le mécanisme d'erreur de votre pilote pour opérer une distinction entre les erreurs transitoires et les erreurs persistantes. Dans le cas d'erreurs transitoires, faites de nouvelles tentatives à l'aide d'un mécanisme de backoff exponentiel, le cas échéant. Assurez-vous que votre application prend en compte la cohérence des données lors de l'implémentation d'une logique de nouvelle tentative.
- Activez la protection contre la suppression de cluster pour tous les clusters de production ou pour tout cluster contenant des données importantes. Avant de supprimer un cluster Amazon DocumentDB, prenez un dernier instantané. Si vous déployez des ressources avec AWS CloudFormation, activez la protection contre le licenciement. Pour plus d'informations, consultez Protection contre la résiliation et la suppression.
- Lors de la création d'un cluster Amazon DocumentDB, le paramètre --engine-version est un paramètre facultatif qui utilise par défaut la dernière version majeure du moteur. La version majeure actuelle du moteur est 4.0.0. Lorsque de nouvelles versions majeures du moteur sont publiées, la version par défaut du moteur pour --engine-version sera mise à jour pour refléter la dernière version du moteur principal. Par conséquent, pour les charges de travail de production, et en particulier celles qui dépendent de scripts, d'automatisation ou de AWS CloudFormation modèles,

nous vous recommandons de spécifier explicitement --engine-version par rapport à la version majeure prévue.

Dimensionnement d'instance

L'un des aspects les plus critiques du choix d'une taille d'instance dans Amazon DocumentDB est la quantité de RAM pour votre cache. Amazon DocumentDB réserve un tiers de la RAM à ses propres services, ce qui signifie que seuls les deux tiers de la RAM de l'instance sont disponibles pour le cache. Il est donc recommandé d'Amazon DocumentDB de choisir un type d'instance disposant de suffisamment de RAM pour adapter votre ensemble de travail (c'est-à-dire les données et les index) en mémoire. Disposer d'instances correctement dimensionnées aide à optimiser les performances globales et à minimiser potentiellement les coûts d'E/S. Vous pouvez utiliser le calculateur de dimensionnement tiers Amazon DocumentDB pour estimer la taille de l'instance pour une charge de travail particulière.

Pour déterminer si la capacité de travail de votre application est suffisante pour la mémoire, surveillez `BufferCacheHitRatio` l'utilisation d'Amazon CloudWatch pour chaque instance d'un cluster en charge.

La `BufferCacheHitRatio` CloudWatch métrique mesure le pourcentage de données et d'index servis à partir du cache mémoire d'une instance (par rapport au volume de stockage). En règle générale, la valeur de `BufferCacheHitRatio` doit être aussi élevée que possible, car la lecture des données à partir de la mémoire de l'ensemble de travail est plus rapide et plus rentable que la lecture à partir du volume de stockage. Bien qu'il soit souhaitable de conserver `BufferCacheHitRatio` le plus proche possible de 100 %, la meilleure valeur possible dépend des modèles d'accès et des exigences de performances de votre application. Pour conserver une valeur la plus élevée possible pour `BufferCacheHitRatio`, il est recommandé que les instances de votre cluster soient provisionnées avec suffisamment de RAM de manière à conserver vos index et vos données de travail en mémoire.

Si vos index ne tiennent pas en mémoire, `BufferCacheHitRatio` aura une valeur moins élevée. La lecture continue à partir du disque entraîne des coûts d'E/S supplémentaires et n'est pas aussi performante que la lecture à partir de la mémoire. Si votre rapport `BufferCacheHitRatio` est moins élevé que prévu, mettez à l'échelle la taille d'instance de votre cluster afin de fournir plus de RAM pour adapter les données de jeu de travail en mémoire. Si la mise à l'échelle de la classe d'instance entraîne une augmentation spectaculaire de `BufferCacheHitRatio`, cela signifie que l'ensemble de travail de votre application ne tenait pas en mémoire. Continuez à monter en

puissance jusqu'à ce que la valeur de `BufferCacheHitRatio` n'augmente plus considérablement après une opération de mise à l'échelle. Pour de plus amples informations sur la surveillance des métriques d'une instance, veuillez consulter Métriques Amazon DocumentDB.

En fonction de vos besoins de charge de travail et de latence, votre application peut avoir des valeurs `BufferCacheHitRatio` plus élevées lors de son utilisation à l'état stable, mais `BufferCacheHitRatio` peut chuter périodiquement lorsque des requêtes analytiques devant analyser une collection entière sont exécutées sur une instance. Ces chutes périodiques de `BufferCacheHitRatio` peuvent se traduire par une latence plus élevée pour les requêtes ultérieures qui doivent repeupler les données de l'ensemble de travail à partir du volume de stockage dans le cache tampon. Nous vous recommandons de tester vos charges de travail dans un environnement de préproduction avec une charge de travail de production représentative afin de comprendre les caractéristiques de performances et **`BufferCacheHitRatio`** avant de déployer la charge de travail en production.

`BufferCacheHitRatio` est une métrique spécifique à l'instance, de sorte que différentes instances d'un même cluster peuvent avoir des valeurs `BufferCacheHitRatio` différentes selon la façon dont les lectures sont réparties entre les instances principale et de réplica. Si votre charge de travail opérationnelle ne peut pas gérer les augmentations périodiques de latence résultant du repeuplement du cache de l'ensemble de travail après l'exécution des requêtes analytiques, essayez d'isoler le cache tampon de la charge de travail normale de celui des requêtes analytiques. Vous pouvez obtenir une isolation `BufferCacheHitRatio` complète en dirigeant les requêtes opérationnelles vers l'instance principale et les requêtes analytiques uniquement vers les instances de réplica. Vous pouvez également réaliser une isolation partielle en dirigeant les requêtes analytiques vers une instance de réplica spécifique, en sachant qu'un certain pourcentage de requêtes régulières s'exécuteront également sur ce réplica et peuvent être affectées.

Les valeurs `BufferCacheHitRatio` appropriées dépendent de votre cas d'utilisation et des exigences de l'application. Il n'y a pas de valeur optimale ou minimale pour cette métrique ; vous seul pouvez décider si le compromis d'une valeur `BufferCacheHitRatio` temporairement inférieure est acceptable du point de vue des coûts et des performances.

Utilisation des index

Création d'index

Lorsque vous importez des données dans Amazon DocumentDB, vous devez créer vos index avant d'importer des ensembles de données volumineux. Vous pouvez utiliser l'outil d'indexation Amazon

DocumentDB pour extraire des index d'une instance ou d'un `mongodump` répertoire MongoDB en cours d'exécution, et créer ces index dans un cluster Amazon DocumentDB. Pour plus d'informations sur les migrations, consultez Migration vers Amazon DocumentDB.

Sélectivité de l'index

Nous vous recommandons de limiter la création d'index aux champs dont le nombre de valeurs en double est inférieur à 1 % du nombre total de documents de la collection. Par exemple, si votre collection contient 100 000 documents, créez uniquement des index sur les champs où la même valeur se produit 1 000 fois ou moins.

Le choix d'un index avec un grand nombre de valeurs uniques (c.-à-d. une cardinalité élevée) garantit que les opérations de filtrage renvoient un petit nombre de documents, ce qui donne de bonnes performances lors des analyses d'index. L'index unique est un exemple d'index de cardinalité élevé, qui garantit que les prédicats d'égalité retournent au plus un seul document. L'index sur un champ booléen et l'index sur le jour de la semaine sont des exemples de faible cardinalité. En raison de leurs performances médiocres, il est peu probable que les index de cardinalité soient choisis par l'optimiseur de requête de la base de données. Dans le même temps, les index de cardinalité faibles continuent de consommer des ressources telles que l'espace disque et les E/S. En règle générale, vous devez cibler les index sur les champs dont la fréquence de valeur type est inférieure ou égale à 1 % de la taille totale de la collection.

En outre, il est recommandé de créer uniquement des index sur les champs qui sont couramment utilisés comme filtre et de rechercher régulièrement des index inutilisés. Pour plus d’informations, consultez Comment analyser l'utilisation des index et identifier les index inutilisés ?.

Incidence des index sur la rédaction de données

Bien que les index puissent améliorer les performances des requêtes en évitant le besoin de numériser tous les documents d'une collection, cette amélioration implique un compromis. Pour chaque index d'une collection, chaque fois qu'un document est inséré, mis à jour ou supprimé, la base de données doit mettre à jour la collection et écrire les champs dans chacun des index de la collection. Par exemple, si une collection comporte neuf index, la base de données doit effectuer dix écritures avant d'accuser réception de l'opération au client. Ainsi, chaque index supplémentaire entraîne une latence d'écriture supplémentaire, des E/S et une augmentation de l'ensemble du stockage utilisé.

Les instances de cluster doivent être dimensionnées de manière appropriée afin de conserver toute la mémoire de l'ensemble de travail. Cela évite de devoir lire en continu les pages d'index à partir du

volume de stockage, ce qui a un impact négatif sur les performances et génère des coûts d'E/S plus élevés. Pour plus d'informations, consultez Dimensionnement d'instance.

Pour de meilleures performances, réduisez le nombre d'index dans vos collections, en ajoutant uniquement les index nécessaires pour améliorer les performances des requêtes courantes. Bien que les charges de travail varient, une bonne recommandation consiste à maintenir le nombre d'index par collection à cinq ou moins.

Identification des index manquants

L'identification des index manquants est une bonne pratique que nous recommandons d'appliquer régulièrement. Pour plus d'informations, consultez Comment identifier les index manquants ?.

Identification des index inutilisés

L'identification et la suppression des index inutilisés est une bonne pratique que nous recommandons d'effectuer régulièrement. Pour plus d'informations, consultez Comment analyser l'utilisation des index et identifier les index inutilisés ?.

Bonnes pratiques de sécurité

Pour respecter les meilleures pratiques en matière de sécurité, vous devez utiliser des comptes AWS Identity and Access Management (IAM) pour contrôler l'accès aux opérations de l'API Amazon DocumentDB, en particulier aux opérations qui créent, modifient ou suppriment des ressources Amazon DocumentDB. Les ressources de ce type incluent les clusters, les groupes de sécurité et les groupes de paramètres. Vous devez également utiliser IAM pour contrôler les actions qui exécutent des actions administratives courantes, telles que la sauvegarde et la restauration de clusters. Lorsque vous créez des rôles IAM, appliquez le principe du moindre privilège.

- Appliquez le principe du moindre privilège avec le contrôle d'accès basé sur les rôles.
- Attribuez un compte IAM individuel à chaque personne qui gère les ressources Amazon DocumentDB. N'utilisez pas l'utilisateur Compte AWS root pour gérer les ressources Amazon DocumentDB. Créez un utilisateur IAM pour chaque personne, y compris vous-même.
- Accordez à chaque utilisateur IAM les autorisations minimales requises pour accomplir ses tâches.
- Utilisez des groupes IAM pour gérer efficacement des autorisations pour plusieurs utilisateurs. Pour de plus amples informations sur IAM, veuillez consulter le Guide de l'utilisateur IAM. Pour de plus amples informations sur les bonnes pratiques IAM, veuillez consulter Bonnes pratiques IAM.

- Effectuez une rotation régulière des informations d'identification IAM.
- Configurez AWS Secrets Manager pour qu'il fasse automatiquement pivoter les secrets pour Amazon DocumentDB. Pour plus d'informations, consultez Rotating Your AWS Secrets Manager secrets et Rotating Secrets for Amazon DocumentDB dans le guide de l'utilisateur de AWS Secrets Manager.
- Accordez à chaque utilisateur Amazon DocumentDB les autorisations minimales requises pour effectuer ses tâches. Pour plus d'informations, consultez Accès à la base de données à l'aide du contrôle d'accès basé sur les rôles.
- Utilisez le protocole TLS (Transport Layer Security) pour chiffrer vos données en transit et AWS KMS pour chiffrer vos données au repos.

Optimisation des coûts

Les meilleures pratiques suivantes peuvent vous aider à gérer et à minimiser vos coûts lorsque vous utilisez Amazon DocumentDB. Pour obtenir des informations sur les tarifs, consultez les FAQ relatives à la tarification d'Amazon DocumentDB (avec compatibilité MongoDB) et les FAQ relatives à Amazon DocumentDB (avec compatibilité MongoDB).

- Créez des alertes de facturation à des seuils de 50 % et 75 % de votre facture prévue pour le mois. Pour plus d'informations sur la création d'alertes de facturation, consultez Création d'une alerte de facturation.
- L'architecture d'Amazon DocumentDB sépare le stockage et le calcul, de sorte que même un cluster à instance unique est très durable. Le volume de stockage de cluster réplique les données six fois sur trois zones de disponibilité, offrant ainsi une durabilité extrêmement élevée, quel que soit le nombre d'instances du cluster. Un cluster de production classique possède trois instances ou plus pour fournir une haute disponibilité. Cependant, vous pouvez optimiser les coûts en utilisant un cluster de développement d'instance unique lorsque la haute disponibilité n'est pas requise.
- Pour les scénarios de développement et de test, arrêtez un cluster lorsqu'il n'est plus nécessaire et démarrez-le lorsque le développement reprend. Pour plus d'informations, consultez Arrêt et démarrage d'un cluster Amazon DocumentDB.
- Les flux TTL et les flux de modification entraînent des E/S lorsque les données sont écrites, lues et supprimées. Si vous avez activé ces fonctionnalités mais que vous ne les utilisez pas dans votre application, vous pouvez les désactiver pour réduire les coûts.

Utilisation des métriques pour identifier les problèmes de performances

Pour identifier les problèmes de performances dus à des ressources insuffisantes ou à d'autres blocages courants, vous pouvez surveiller les métriques disponibles pour votre cluster Amazon DocumentDB.

Consultation des métriques de performances

Vous devez régulièrement surveiller les métriques de performances pour observer les valeurs moyennes, maximales et minimales à différents intervalles de temps. Cela vous aide à déterminer quand les performances se dégradent. Vous pouvez également définir des CloudWatch alarmes Amazon pour des seuils métriques spécifiques afin d'être alerté s'ils sont atteints.

Pour résoudre les problèmes de performances, il est important de comprendre les performances de base du système. Lorsque vous configurez un nouveau cluster et l'exécutez avec une charge de travail typique, capturez les valeurs moyennes, maximum et minimum de toutes les métriques de performances à différents intervalles (par exemple, une heure, 24 heures, une semaine, deux semaines). Cela vous permet de vous faire une idée de ce qui est normal. Cela permet de comparer l'activité pendant les heures pleines et les heures creuses. Vous pouvez ensuite utiliser ces informations pour identifier quand les performances chutent sous les niveaux standard.

Vous pouvez consulter les indicateurs de performance à l'aide du AWS Management Console ou AWS CLI. Pour plus d’informations, consultez Visualisation CloudWatch Données.

Configuration d'une CloudWatch alarme

Pour définir une CloudWatch alarme, consultez la section Utilisation d'Amazon CloudWatch Alarms dans le guide de CloudWatch l'utilisateur Amazon.

Evaluation des métriques de performances

Une instance possède différentes catégories de métriques. La façon de déterminer les valeurs acceptables dépend de la métrique.

CPU

- Utilisation du processeur : pourcentage de la capacité de traitement de l'ordinateur utilisée.

Mémoire

- Mémoire disponible : quantité de RAM disponible sur l'instance.
- Utilisation du swap : quantité d'espace de swap utilisée par l'instance, en mégaoctets.

Opérations d'entrée/sortie

- IOPS en lecture, IOPS en écriture : nombre moyen d'opérations de lecture ou d'écriture sur le disque par seconde.
- Latence de lecture, latence d'écriture : durée moyenne d'une opération de lecture ou d'écriture en millisecondes.
- Débit de lecture, débit d'écriture : nombre moyen de mégaoctets lus ou écrits sur le disque par seconde.
- Profondeur de la file d'attente du disque : nombre d'opérations d'E/S en attente d'écriture ou de lecture sur le disque.

Trafic réseau

- Débit de réception réseau, débit de transmission réseau : débit du trafic réseau à destination et en provenance de l'instance en mégaoctets par seconde.

Connexions de la base de données

- Connexions à la base de données : nombre de sessions client connectées à l'instance.

En général, les valeurs acceptables pour les métriques de performances dépendent de vos données de référence et de l'activité de votre application. Enquêtez sur les écarts cohérents ou tendanciels de vos données de référence.

Voici quelques recommandations et conseils sur les types spécifiques de métriques :

- Consommation élevée du processeur : des valeurs élevées de consommation du processeur peuvent être appropriées, à condition qu'elles soient conformes aux objectifs de votre application (tels que le débit ou la simultanéité) et qu'elles soient attendues. Si votre consommation d'UC est constamment supérieure à 80 %, pensez à augmenter la capacité de vos instances.

- Consommation de RAM élevée : si votre `FreeableMemory` indicateur tombe fréquemment en dessous de 10 % de la mémoire totale de l'instance, pensez à augmenter le volume de vos instances. Pour plus d'informations sur ce qui se passe lorsque votre instance DocumentDB est confrontée à une charge de mémoire élevée, consultez Amazon DocumentDB Resource Governance.
- Utilisation des swaps — Cette métrique doit rester égale ou proche de zéro. Si votre utilisation de l'échange est importante, envisagez de dimensionner vos instances.
- Trafic réseau : pour le trafic réseau, contactez votre administrateur système afin de connaître le débit attendu pour le réseau de votre domaine et votre connexion Internet. Enquêtez sur le trafic réseau si le débit est constamment inférieur à vos attentes.
- Connexions aux bases de données — Envisagez de restreindre les connexions aux bases de données si vous constatez un nombre élevé de connexions utilisateur ainsi qu'une baisse des performances des instances et du temps de réponse. Le bon nombre de connexions utilisateur pour votre instance de base de données dépend de votre classe d'instance et de la complexité des opérations exécutées. Si vous rencontrez des problèmes avec les métriques de performances, l'une des premières choses à faire pour améliorer les choses est de régler les requêtes les plus utilisées et onéreuses pour réduire la pression exercée sur les ressources système.

Si vos requêtes sont réglées et que le problème persiste, envisagez de mettre à niveau votre classe d'instance Amazon DocumentDB vers une classe contenant davantage de ressources (processeur, RAM, espace disque, bande passante réseau, capacité d'E/S) associées au problème que vous rencontrez.

Réglage des requêtes

L'un des meilleurs moyens d'améliorer les performances d'un cluster consiste à régler les requêtes les plus communément utilisées et exigeantes en ressources pour les rendre moins onéreuses à exécuter.

Vous pouvez utiliser le profileur (voir Profilage des opérations Amazon DocumentDB) pour enregistrer l'heure d'exécution et les détails des opérations qui ont été effectuées sur votre cluster. Le profileur est utile pour surveiller les opérations les plus lentes sur votre cluster afin de vous aider à améliorer les performances des requêtes individuelles et les performances globales du cluster.

Vous pouvez également utiliser la commande `explain` pour apprendre à analyser un plan de requête pour une requête particulière. Vous pouvez utiliser ces informations pour modifier une

requête ou collection sous-jacente afin d'améliorer les performances de vos requêtes (par exemple, en ajoutant un index).

Charges de travail TTL et en séries chronologiques

La suppression de documents résultant de l'expiration de l'index TTL est un processus qui demande un effort maximal. Il n’est pas certain que les documents soient supprimés dans un délai spécifique. Des facteurs tels que la taille de l'instance, l'utilisation des ressources de l'instance, la taille du document, le débit global, le nombre d'index et l'adéquation des index et du jeu de travail dans la mémoire peuvent tous avoir une incidence sur le moment où les documents expirés sont supprimés par le processus TTL.

Lorsque le moniteur TTL supprime vos documents, chaque suppression entraîne des coûts d'E/S, ce qui augmente le montant de votre facture. Si le débit et les taux de suppression TTL augmentent, vous devez vous attendre à une facture plus élevée en raison de l'utilisation accrue des E/S. Toutefois, si vous ne créez pas d'index TTL pour supprimer des documents, mais que vous segmentez les documents en collections en fonction du temps et que vous supprimez simplement ces collections lorsqu'elles ne sont plus nécessaires, vous n'aurez aucun coût d'E/S à payer. Cela peut être nettement plus rentable que l'utilisation d'un indice TTL.

Pour les charges de travail en séries chronologiques, vous pouvez envisager de créer des collections continues au lieu d'un index TTL, car les collections continues peuvent constituer un moyen plus performant de supprimer des données tout en étant moins gourmand en E/S. Si vous avez des collections volumineuses (en particulier des collections supérieures à 1 To) ou si les coûts d'E/S de suppression de TTL sont élevés, nous vous recommandons de partitionner les documents dans des collections en fonction du temps et de supprimer les collections lorsque les documents ne sont plus nécessaires. Vous pouvez créer une collection par jour ou une par semaine, en fonction de votre taux d'ingestion de données. Bien que les exigences varient en fonction de votre application, une bonne règle consiste à avoir davantage de petites collections plutôt que quelques grandes collections. La suppression de ces collections n'entraîne pas de coûts d'E/S et peut s'avérer plus rapide et plus rentable que l’utilisation d’un index TTL.

Migrations

En tant que bonne pratique, nous vous recommandons, lors de la migration de données vers Amazon DocumentDB, de créer d'abord vos index dans Amazon DocumentDB avant de migrer les données. La création d'index d'abord peut réduire le temps global et augmenter la vitesse de la migration. Pour

ce faire, vous pouvez utiliser l'outil d'indexation Amazon DocumentDB. Pour plus d'informations sur les migrations, consultez le guide de migration Amazon DocumentDB.

Avant de migrer votre base de données de production, nous vous recommandons également de tester entièrement votre application sur Amazon DocumentDB, en tenant compte des fonctionnalités, des performances, des opérations et des coûts.

Utilisation des groupes de paramètres de cluster

Nous vous recommandons de tester les modifications apportées aux groupes de paramètres de cluster sur un cluster test avant d'appliquer ces modifications à vos clusters de production. Pour de plus amples informations sur la sauvegarde de votre cluster, veuillez consulter Sauvegarde et restauration dans Amazon DocumentDB.

Requêtes de pipeline d'agrégation

Lors de la création d'une requête de pipeline d'agrégation avec plusieurs étapes et de l'évaluation d'un sous-ensemble de données dans la requête, utilisez l'étape `$match` comme première étape ou au début du pipeline. L'utilisation de `$match` en premier permet de réduire le nombre de documents que les étapes suivantes de la requête de pipeline d'agrégation devront traiter. Les performances de votre requête en seront améliorées.

`batchInsert` et `batchUpdate`

Lorsque vous effectuez un taux élevé de simultanéité `batchInsert` et/ou d'`batchUpdate`opérations et que le montant de `FreeableMemory` (CloudWatch métrique) passe à zéro sur votre instance principale, vous pouvez soit réduire la simultanéité de l'insertion par lots, soit mettre à jour la charge de travail, soit, si la simultanéité de la charge de travail ne peut pas être réduite, augmenter la taille de l'instance pour augmenter la quantité de. `FreeableMemory`

Différences fonctionnelles : Amazon DocumentDB et MongoDB

Voici les différences fonctionnelles entre Amazon DocumentDB (compatible avec MongoDB) et MongoDB.

Rubriques

Avantages fonctionnels d'Amazon DocumentDB

Transactions implicites

Dans Amazon DocumentDB, toutes les instructions CRUD (`findAndModify`,, `updateinsert`,`delete`) garantissent l'atomicité et la cohérence, même pour les opérations qui modifient plusieurs documents. Avec le lancement d'Amazon DocumentDB 4.0, les transactions explicites fournissant des propriétés ACID pour les opérations multi-instructions et multi-collections sont désormais prises en charge. Pour en savoir plus sur l'utilisation des transactions dans Amazon DocumentDB, consultez. Transactions

Vous trouverez ci-dessous des exemples d'opérations dans Amazon DocumentDB qui modifient plusieurs documents répondant à la fois à des comportements atomiques et cohérents.

```
db.miles.update(
    { "credit_card": { $eq: true } },
    { $mul: { "flight_miles.$[]": NumberInt(2) } },
    { multi: true }
)
```

```
db.miles.updateMany(
    { "credit_card": { $eq: true } },
    { $mul: { "flight_miles.$[]": NumberInt(2) } }
```

```
)
```

```
db.runCommand({
  update: "miles",
  updates: [
    {
      q: { "credit_card": { $eq: true } },
      u: { $mul: { "flight_miles.$[]": NumberInt(2) } },
      multi: true
    }
  ]
})
```

```
db.products.deleteMany({
  "cost": { $gt: 30.00 }
})
```

```
db.runCommand({
  delete: "products",
  deletes: [{ q: { "cost": { $gt: 30.00 } }, limit: 0 }]
})
```

Les opérations individuelles qui composent les opérations en bloc comme `updateMany` et `deleteMany` sont atomiques. Toutefois, cela ne signifie pas que les opérations en vrac sont entièrement atomiques. Par exemple, l'intégralité de l'opération `insertMany` est atomique si les opérations d'insertion individuelles s'exécutent avec succès sans erreur. En cas d'erreur lors d'une `insertMany` opération, chaque instruction d'insertion individuelle contenue dans l'`insertMany`opération sera exécutée comme une opération atomique. Si vous avez besoin de propriétés ACID pour les `insertMany` `deleteMany` opérations et les opérations, il est recommandé d'utiliser une transaction. `updateMany`

Différences fonctionnelles mises à jour

Amazon DocumentDB continue d'améliorer la compatibilité avec MongoDB en remontant les fonctionnalités que nos clients nous demandent de développer. Cette section contient les différences

fonctionnelles que nous avons supprimées dans Amazon DocumentDB afin de faciliter les migrations et la création d'applications pour nos clients.

Rubriques

Indexation de tableau

Depuis le 23 avril 2020, Amazon DocumentDB permet désormais d'indexer des tableaux de plus de 2 048 octets. La limite pour un élément individuel dans un tableau reste de 2 048 octets, ce qui est cohérent avec MongoDB.

Si vous créez un nouvel index, aucune action n'est nécessaire pour profiter de la fonctionnalité améliorée. Si vous avez un index existant, vous pouvez profiter de la fonctionnalité améliorée en l'abandonnant, puis en le recréant. La version d'index actuelle avec les capacités améliorées est `"v" : 3`.

> **Note**
>
> Pour les clusters de production, la suppression de l'index peut avoir un impact sur les performances de votre application. Nous vous recommandons d'abord de tester et de procéder avec prudence lorsque vous apportez des modifications à un système de production. De plus, le temps qu'il faudra pour recréer l'index dépendra de la taille globale des données de la collection.

Vous pouvez interroger la version de vos index à l'aide de la commande suivante.

```
db.collection.getIndexes()
```

Le résultat de cette opération ressemble à ceci. Dans cette sortie, la version de l'index est `"v" : 3`, qui est la version d'index la plus récente.

```
[
    {
        "v" : 3,
        "key" : {
        "_id" : 1
        },
        "name" : "_id_",
        "ns" : "test.test"
    }
]
```

Index multiclés

Depuis le 23 avril 2020, Amazon DocumentDB permet désormais de créer un index composé avec plusieurs clés dans le même tableau.

Si vous créez un nouvel index, aucune action n'est nécessaire pour profiter de la fonctionnalité améliorée. Si vous avez un index existant, vous pouvez profiter de la fonctionnalité améliorée en l'abandonnant, puis en le recréant. La version d'index actuelle avec les capacités améliorées est `"v" : 3`.

> **Note**
>
> Pour les clusters de production, la suppression de l'index peut avoir un impact sur les performances de votre application. Nous vous recommandons d'abord de tester et de procéder avec prudence lorsque vous apportez des modifications à un système de production. De plus, le temps qu'il faudra pour recréer l'index dépendra de la taille globale des données de la collection.

Vous pouvez interroger la version de vos index à l'aide de la commande suivante.

```
db.collection.getIndexes()
```

Le résultat de cette opération ressemble à ceci. Dans cette sortie, la version de l'index est `"v" : 3`, qui est la version d'index la plus récente.

```
[
    {
```

```
        "v" : 3,
        "key" : {
            "_id" : 1
        },
        "name" : "_id_",
        "ns" : "test.test"
    }
]
```

Caractères null dans les chaînes

Depuis le 22 juin 2020, Amazon DocumentDB prend désormais en charge les caractères nuls (`'\0'`) dans les chaînes.

Contrôle d'accès basé sur les rôles

Depuis le 26 mars 2020, Amazon DocumentDB prend en charge le contrôle d'accès basé sur les rôles (RBAC) pour les rôles intégrés. Pour en savoir plus, veuillez consulter la section Contrôle d'accès basé sur les rôles.

Indexation **`$regex`**

Depuis le 22 juin 2020, Amazon DocumentDB permet désormais aux `$regex` opérateurs d'utiliser un index.

Pour utiliser un index avec l'opérateur `$regex`, vous devez utiliser la commande `hint()`. Lorsque vous utilisez `hint()`, vous devez spécifier le nom du champ auquel vous appliquez le `$regex`. Par exemple, si vous avez un index sur le champ `product` avec le nom d'index `p_1`, `db.foo.find({product: /^x.*/}).hint({product:1})` utilisera l'index `p_1`, mais `db.foo.find({product: /^x.*/}).hint("p_1")` n'utilisera pas l'index. Vous pouvez vérifier si un index est choisi à l'aide de la commande `explain()` ou du profileur pour consigner les requêtes lentes. Par exemple, `db.foo.find({product: /^x.*/}).hint("p_1").explain()`.

> **Note**
>
> La méthode `hint()` ne peut être utilisée qu'avec un index à la fois.

L'utilisation d'un index pour une requête `$regex` est optimisée pour les requêtes regex qui utilisent un préfixe et ne spécifient pas les options regex `I`, `m` ou `o`.

Lorsque vous utilisez un index avec `$regex`, il est recommandé de créer un index sur des champs hautement sélectifs où le nombre de valeurs en double est inférieur à 1 % du nombre total de documents de la collection. Par exemple, si votre collection contient 100 000 documents, créez uniquement des index sur les champs où la même valeur se produit 1 000 fois ou moins.

Projection pour les documents imbriqués

Il existe une différence fonctionnelle d'`$project`opérateur entre Amazon DocumentDB et MongoDB dans la version 3.6 qui a été résolue dans Amazon DocumentDB 4.0 mais ne sera toujours pas prise en charge dans Amazon DocumentDB 3.6.

Amazon DocumentDB 3.6 ne prend en compte que le premier champ d'un document imbriqué lors de l'application d'une projection, tandis que MongoDB 3.6 analyse les sous-documents et applique également la projection à chaque sous-document.

Par exemple : si la projection est le cas`"a.b.c": 1`, le comportement fonctionne comme prévu dans Amazon DocumentDB et MongoDB. Toutefois, si la projection est le cas`{a:{b:{c:1}}}`, Amazon DocumentDB 3.6 appliquera uniquement la projection à `a` et non `b` à ou. `c` Dans Amazon DocumentDB 4.0, la projection `{a:{b:{c:1}}}` sera appliquée à `ab`, et. `c`

Différences fonctionnelles avec MongoDB

Rubriques

- Opérateur $vectorSearch
- OpCountersCommand
- Bases de données et collections d'administration
- cursormaxTimeMS
- explain()
- Restrictions relatives aux noms de champ
- Générations d'index
- Recherche avec une clé vide dans le chemin
- API MongoDB, opérations et types de données
- mongodumpet mongorestore services publics

- Ordre des résultats
- Écritures réessayables
- Index fragmenté
- Utilisation de l'élément $elemMatch dans une expression $all
- $ne,$nin, nornot,$exists, et $elemMatch indexation
- $lookup

Opérateur **$vectorSearch**

Amazon DocumentDB n'est pas compatible en `$vectorSearch` tant qu'opérateur indépendant. Au lieu de cela, nous soutenons, au `vectorSearch` sein de l'`$search`opérateur. Pour plus d'informations, consultez Recherche vectorielle pour Amazon DocumentDB.

`OpCountersCommand`

Le `OpCountersCommand` comportement d'Amazon DocumentDB diffère de celui de MongoDB comme suit : `opcounters.command`

- MongoDB `opcounters.command` compte toutes les commandes à l'exception de l'insertion, de la mise à jour et de la suppression, tandis que Amazon DocumentDB exclut `OpCountersCommand` également la `find` commande.
- Amazon DocumentDB compte les commandes internes (telles que`getCloudWatchMetricsV2`) vers. `OpCountersCommand`

Bases de données et collections d'administration

Amazon DocumentDB ne prend pas en charge l'administration ou la base de données locale, ni MongoDB `system.*` ou les collections respectivement. `startup_log`

`cursormaxTimeMS`

Dans Amazon DocumentDB, `cursor.maxTimeMS` réinitialise le compteur pour chaque demande. `getMore` Ainsi, si une valeur de 3 000 MS `maxTimeMS` est spécifiée, que la requête prend 2 800 ms et que chaque `getMore` demande suivante prend 300 MS, le curseur n'expirera pas. Le curseur n'expire que lorsqu'une seule opération, qu'il s'agisse de la requête ou d'une `getMore` demande

individuelle, prend plus que ce qui est spécifié`maxTimeMS`. De plus, le balayeur qui vérifie le temps d'exécution du curseur fonctionne à une granularité de cinq (5) minutes.

explain()

Amazon DocumentDB émule l'API MongoDB 4.0 sur un moteur de base de données spécialement conçu qui utilise un système de stockage distribué, tolérant aux pannes et autoréparateur. Par conséquent, les plans de requête et le résultat de `explain()` peuvent différer entre Amazon DocumentDB et MongoDB. Les clients qui souhaitent contrôler leur plan de requête peuvent utiliser l'opérateur `$hint` pour appliquer la sélection d'un index préféré.

Restrictions relatives aux noms de champ

Amazon DocumentDB ne prend pas en charge les points « ». dans le nom d'un champ de document, par exemple,`db.foo.insert({‘x.1’:1})`.

Amazon DocumentDB ne prend pas non plus en charge le préfixe $ dans les noms de champs.

Par exemple, essayez la commande suivante dans Amazon DocumentDB ou MongoDB :

```
rs0:PRIMARY< db.foo.insert({"a":{"$a":1}})
```

MongoDB renverra ce qui suit :

```
WriteResult({ "nInserted" : 1 })
```

Amazon DocumentDB renverra une erreur :

```
WriteResult({
"nInserted" : 0,
"writeError" : {
"code" : 2,
"errmsg" : "Document can't have $ prefix field names: $a"
}
})
```

> **Note**
>
> Il existe une exception à cette différence fonctionnelle. Les noms de champs suivants commençant par le préfixe $ ont été ajoutés à la liste blanche et peuvent être utilisés avec succès dans Amazon DocumentDB : $id, $ref et $db.

Générations d'index

Amazon DocumentDB autorise la création d'un seul index sur une collection à la fois. Au premier plan ou en arrière-plan. Si des opérations telles que `createIndex()` ou `dropIndex()` se produisent sur la même collection lorsqu'une génération d'index est en cours, l'opération nouvellement tentée échoue.

Par défaut, les compilations d'index dans Amazon DocumentDB et MongoDB version 4.0 s'effectuent en arrière-plan. MongoDB version 4.2 et versions ultérieures ignorent l'option de création d'index d'arrière-plan si elle est spécifiée à CreateIndexes ou à ses assistants shell et. `createIndex()` `createIndexes()`

Un index Time to Live (TTL) commence à expirer les documents une fois la création de l'index terminée.

Recherche avec une clé vide dans le chemin

Lorsque vous recherchez une clé qui inclut une chaîne vide dans le chemin (par exemple`x.,x..b`) et que l'objet possède un chemin de clé de chaîne vide (par exemple`{"x" : [ { "" : 10 }, { "b" : 20 } ]}`) dans un tableau, Amazon DocumentDB renvoie des résultats différents de ceux que vous obtiendriez si vous exécutiez la même recherche dans MongoDB.

Dans MongoDB, la recherche du chemin de clé vide dans le tableau fonctionne comme prévu lorsque la clé de chaîne vide ne se trouve pas à la fin de la recherche de chemin. Cependant, lorsque la clé de chaîne vide se trouve à la fin de la recherche du chemin, elle n'apparaît pas dans le tableau.

Cependant, dans Amazon DocumentDB, seul le premier élément du tableau est lu, car il `getArrayIndexFromKeyString` convertit une chaîne vide en chaîne. La recherche par clé de chaîne est donc traitée comme une recherche d'index de tableau. `0`

API MongoDB, opérations et types de données

Amazon DocumentDB est compatible avec les API MongoDB 3.6 et 4.0. Pour obtenir la up-to-date liste des fonctionnalités prises en charge, consultezAPI MongoDB, opérations et types de données pris en charge.

`mongodump`et `mongorestore` services publics

Amazon DocumentDB ne prend pas en charge une base de données d'administration et ne vide donc ni ne restaure la base de données d'administration lors de l'utilisation des utilitaires `mongodump` or`mongorestore`. Lorsque vous créez une nouvelle base de données dans Amazon DocumentDB à l'aide de`mongorestore`, vous devez recréer les rôles utilisateur en plus de l'opération de restauration.

> **Note**
>
> Nous recommandons les outils de base de données MongoDB jusqu'à la version 100.6.1 incluse pour Amazon DocumentDB. Vous pouvez accéder aux téléchargements des outils de base de données MongoDB ici.

Ordre des résultats

Amazon DocumentDB ne garantit pas l'ordre de tri implicite des ensembles de résultats. Pour garantir l'ordre d'un jeu de résultats, spécifiez explicitement un ordre de tri en utilisant `sort()`.

L'exemple suivant trie les éléments de la collecte d'inventaire par ordre décroissant en fonction du champ stock.

```
db.inventory.find().sort({ stock: -1 })
```

Lors de l'utilisation de l'étape d'`$sort`agrégation, l'ordre de tri n'est pas préservé, sauf si l'`$sort`étape est la dernière étape du pipeline d'agrégation. Lorsque l'étape d'`$sort`agrégation est utilisée en combinaison avec la phase d'`$group``$sort`agrégation, l'étape d'agrégation est uniquement appliquée aux `$last` accumulateurs `$first` et. Dans Amazon DocumentDB 4.0, la prise en charge du respect de l'ordre de tri par rapport `$push` à l'étape précédente `$sort` a été ajoutée.

Écritures réessayables

À partir des pilotes compatibles MongoDB 4.2, les écritures réessayables sont activées par défaut. Cependant, Amazon DocumentDB ne prend actuellement pas en charge les écritures réessayables. La différence fonctionnelle se manifeste dans un message d'erreur similaire à ce qui suit.

```
{"ok":0,"errmsg":"Unrecognized field: 'txnNumber'","code":9,"name":"MongoError"}
```

Les écritures réessayables peuvent être désactivées via la chaîne de connexion (par exemple, `MongoClient("mongodb://my.mongodb.cluster/db?retryWrites=false"))` ou l'argument mot-clé du MongoClient constructeur (par exemple,. `MongoClient("mongodb://my.mongodb.cluster/db", retryWrites=False))`

Voici un exemple Python qui désactive les écritures réessayables dans la chaîne de connexion.

```
client =
 pymongo.MongoClient('mongodb://
<username>:<password>@docdb-2019-03-17-16-49-12.cluster-ccuszbx3pn5e.us-
east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0',w='majority',j=True,retryWrites=False)
```

Index fragmenté

Pour utiliser un index fragmenté que vous avez créée dans une requête, vous devez utiliser la clause `$exists` sur les champs qui couvrent l'index. Si vous omettez`$exists`, Amazon DocumentDB n'utilise pas l'index clairsemé.

Voici un exemple.

```
db.inventory.count({ "stock": { $exists: true }})
```

Pour les index multiclés épars, Amazon DocumentDB ne prend pas en charge une contrainte de clé unique si la recherche d'un document aboutit à un ensemble de valeurs et que seul un sous-ensemble des champs indexés est manquant. Par exemple, `createIndex({"a.b" : 1 }, { unique : true, sparse :true })` n'est pas pris en charge, étant donné l'entrée de `"a" : [ { "b" : 2 }, { "c" : 1 } ]`, car `"a.c"` est stocké dans l'index.

Utilisation de l'élément $elemMatch dans une expression $all

Amazon DocumentDB ne prend actuellement pas en charge l'utilisation de l'`$elemMatch`opérateur dans une `$all` expression. Comme solution de contournement, vous pouvez utiliser l'opérateur `$and` avec `$elemMatch` comme suit.

Opération d'origine :

```
db.col.find({
  qty: {
    $all: [
      { "$elemMatch": { part: "xyz", qty: { $lt: 11 } } },
      { "$elemMatch": { num: 40, size: "XL" } }
    ]
  }
})
```

Opération mise à jour :

```
db.col.find({
  $and: [
    { qty: { "$elemMatch": { part: "xyz", qty: { $lt: 11 } } } },
    { qty: { "$elemMatch": { qty: 40, size: "XL" } } }
  ]
})
```

`$ne`,**`$nin`**, **`$nor$not`**,**`$exists`**, et **`$elemMatch`** indexation

Amazon DocumentDB ne prend actuellement pas en charge la possibilité d'utiliser des index avec les opérateurs`$ne`,`$nin`,`$nor`, `$not$exists`, et. `$distinct` Par conséquent, l'utilisation de ces opérateurs entraînera des scans des collections. L'exécution d'un filtre ou d'une correspondance avant d'utiliser l'un de ces opérateurs permet de réduire la quantité de données à scanner et d'améliorer ainsi les performances.

Amazon DocumentDB a ajouté la prise en charge des scans d'index avec l'`$elemMatch`opérateur dans Amazon DocumentDB 5.0 et des clusters élastiques. Les analyses d'index sont prises en charge lorsque le filtre réservé aux requêtes possède un niveau de `$elemMatch` filtre, mais elles ne sont pas prises en charge si une `$elemMatch` requête imbriquée est incluse.

`$elemMatch`forme de requête qui prend en charge les analyses d'index dans Amazon DocumentDB 5.0 :

```
db.foo.find( { "a": {$elemMatch: { "b": "xyz", "c": "abc"} } })
```

`$elemMatch`forme de requête qui ne prend pas en charge les analyses d'index dans Amazon DocumentDB 5.0 :

```
db.foo.find( { "a": {$elemMatch: { "b": {$elemMatch: { "d": "xyz", "e": "abc"} }} } })
```

$lookup

Amazon DocumentDB permet d'effectuer des correspondances d'égalité (par exemple, jointure externe gauche) et prend également en charge les sous-requêtes non corrélées, mais ne prend pas en charge les sous-requêtes corrélées.

Utilisation d'un index avec $lookup

Vous pouvez désormais utiliser un index avec l'opérateur `$lookup` stage. Selon votre cas d'utilisation, il existe plusieurs algorithmes d'indexation que vous pouvez utiliser pour optimiser les performances. Cette section explique les différents algorithmes d'indexation `$lookup` et vous aide à choisir celui qui convient le mieux à votre charge de travail.

Par défaut, Amazon DocumentDB utilise l'algorithme de hachage lorsqu'il `allowDiskUse:false` est utilisé et la fusion de tri lorsqu'il `allowDiskUse:true` est utilisé. Dans certains cas d'utilisation, il peut être souhaitable de forcer l'optimiseur de requêtes à utiliser un algorithme différent. Vous trouverez ci-dessous les différents algorithmes d'indexation que l'opérateur d'`$lookup`agrégation peut utiliser :

- Boucle imbriquée : un plan de boucle imbriquée est généralement avantageux pour une charge de travail si la collection étrangère est inférieure à 1 Go et si le champ de la collection étrangère possède un index. Si l'algorithme de boucle imbriquée est utilisé, le plan d'explication indiquera la scène sous `NESTED_LOOP_LOOKUP` la forme.
- Fusion de tri : un plan de fusion de tri est généralement avantageux pour une charge de travail si la collection étrangère ne possède pas d'index sur le champ utilisé pour la recherche et si le jeu de données de travail ne tient pas dans la mémoire. Si l'algorithme de fusion de tri est utilisé, le plan d'explication indiquera l'étape sous la forme`SORT_LOOKUP`.

- Hachage : un plan de hachage est généralement avantageux pour une charge de travail si la collection étrangère est inférieure à 1 Go et si le jeu de données de travail est conservé en mémoire. Si l'algorithme de hachage est utilisé, le plan d'explication indiquera l'étape sous HASH_LOOKUP la forme.

Vous pouvez identifier l'algorithme d'indexation utilisé pour l'`$lookup`opérateur en utilisant la commande explain dans la requête. Vous trouverez ci-dessous un exemple.

```
db.localCollection.explain().
aggregate( [

   {
          $lookup:
                   {
                             from: "foreignCollection",
                             localField: "a",
       foreignField: "b",
       as: "joined"
}
   }
]

output
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "test.localCollection",
        "winningPlan" : {
            "stage" : "SUBSCAN",
            "inputStage" : {
                "stage" : "SORT_AGGREGATE",
                "inputStage" : {
                    "stage" : "SORT",
                    "inputStage" : {
                        "stage" : "NESTED_LOOP_LOOKUP",
                        "inputStages" : [
                            {
                                "stage" : "COLLSCAN"
                            },
                            {
                                "stage" : "FETCH",
                                "inputStage" : {
```

```
                                    "stage" : "COLLSCAN"
                                }
                            }
                        ]
                    }
                }
            }
        }
    },
    "serverInfo" : {
        "host" : "devbox-test",
        "port" : 27317,
        "version" : "3.6.0"
    },
    "ok" : 1
}
```

Au lieu d'utiliser la `explain()` méthode, vous pouvez utiliser le profileur pour examiner l'algorithme utilisé lors de votre utilisation de l'`$lookup`opérateur. Pour plus d'informations sur le profileur, veuillez consulterProfilage des opérations Amazon DocumentDB.

Utilisation d'un **`planHint`**

Si vous souhaitez forcer l'optimiseur de requêtes à utiliser un algorithme d'indexation différent`$lookup`, vous pouvez utiliser un. `planHint` Pour ce faire, utilisez le commentaire dans les options de la phase d'agrégation pour forcer un plan différent. Voici un exemple de syntaxe du commentaire :

```
comment : {
    comment :  “<string>”,
    lookupStage : { planHint : “SORT” | “HASH” | "NESTED_LOOP" }
}
```

Vous trouverez ci-dessous un exemple d'utilisation de `planHint` pour forcer l'optimiseur de requêtes à utiliser l'algorithme d'HASHindexation :

```
db.foo.aggregate(
  [
    {
        $lookup:
        {
```

```
            from: "foo",
            localField: "_id",
            foreignField: "_id",
            as: "joined"
        },
    }
  ],
  {
    comment : "{ \\"lookupStage\\" : { \\"planHint\\": \\"HASH\\" }}"
```

Pour tester l'algorithme le mieux adapté à votre charge de travail, vous pouvez utiliser le `executionStats` paramètre de la `explain` méthode pour mesurer le temps d'exécution de l'`$lookup`étape tout en modifiant l'algorithme d'indexation (c'est-à-dire`HASH`/`SORT`/`NESTED_LOOP`).

L'exemple suivant montre comment mesurer le temps `executionStats` d'exécution de l'`$lookup`étape à l'aide de l'`SORT`algorithme.

```
db.foo.explain(“executionStats”).aggregate(
  [
    {
        $lookup:
        {
            from: "foo",
            localField: "_id",
            foreignField: "_id",
            as: "joined"
        },
    }
  ],
  {
    comment : "{ \\"lookupStage\\" : { \\"planHint\": \\"SORT\\" }}"
```

API MongoDB, opérations et types de données pris en charge

Amazon DocumentDB (compatible avec MongoDB) est un service de base de données de documents rapide, évolutif, hautement disponible et entièrement géré qui prend en charge les charges de travail MongoDB. Amazon DocumentDB est compatible avec les API MongoDB 3.6, 4.0 et 5.0. Cette section répertorie les fonctionnalités prises en charge. Pour obtenir de l'aide sur l'utilisation des API et des pilotes MongoDB, veuillez consulter les forums de la communauté MongoDB. Pour obtenir de l'aide sur le service Amazon DocumentDB, contactez l'équipe d' AWS assistance appropriée. Pour connaître les différences fonctionnelles entre Amazon DocumentDB et MongoDB, consultez. Différences fonctionnelles : Amazon DocumentDB et MongoDB

Les commandes et opérateurs MongoDB internes uniquement ou non applicables à un service entièrement géré ne sont pas pris en charge et ne sont pas inclus dans la liste des fonctionnalités prises en charge.

Depuis le lancement, nous avons ajouté plus de 50 fonctions supplémentaires et nous continuerons à prendre en compte les retours de nos clients pour fournir les fonctions dont ils ont besoin. Pour plus d'informations sur les derniers lancements, consultez les annonces d'Amazon DocumentDB.

Si vous souhaitez que nous développions une fonctionnalité qui n'est pas prise en charge, veuillez nous en informer en envoyant un e-mail avec votre AccountID, les fonctionnalités demandées et le cas d'utilisation à l'équipe du service Amazon DocumentDB.

Rubriques

Commandes de base de données

Rubriques

Commandes administratives

Command	3.6	4.0	5.0	Cluster élastique
Collections limitées	Non	Non	Non	Non
clone : CollectionAs Capuché	Non	Non	Non	Non
collMod	Partielle	Partielle	Partielle	Partielle
CollMod : expireAfterSeconds	Oui	Oui	Oui	Oui
convertirToCapped	Non	Non	Non	Non
copydb	Non	Non	Non	Non
créer	Oui	Oui	Oui	Oui

Command	3.6	4.0	5.0	Cluster élastique
createView	Non	Non	Non	Non
createIndexes	Oui	Oui	Oui	Oui
currentOp	Oui	Oui	Oui	Oui
drop	Oui	Oui	Oui	Oui
dropDatabase	Oui	Oui	Oui	Oui
dropIndexes	Oui	Oui	Oui	Oui
filemd5	Non	Non	Non	Non
killCursors	Oui	Oui	Oui	Oui
killOp	Oui	Oui	Oui	Oui
Liste des collections*	Oui	Oui	Oui	Oui
listDatabases	Oui	Oui	Oui	Oui
listIndexes	Oui	Oui	Oui	Oui
reIndex	Non	Non	Non	Non
renameCollection	Oui	Oui	Oui	Non

* La `type` touche de l'option de filtre n'est pas prise en charge.

Agrégation

Command	3.6	4.0	5.0	Cluster élastique
aggregate	Oui	Oui	Oui	Oui

Command	3.6	4.0	5.0	Cluster élastique
count	Oui	Oui	Oui	Oui
distinct	Oui	Oui	Oui	Oui
mapReduce	Non	Non	Non	Non

Authentification

Command	3.6	4.0	5.0	Cluster élastique
authenticate	Oui	Oui	Oui	Oui
logout	Oui	Oui	Oui	Oui

Commandes de diagnostic

Command	3.6	4.0	5.0	Cluster élastique
buildInfo	Oui	Oui	Oui	Oui
collStats	Oui	Oui	Oui	Oui
conn PoolStats	Non	Non	Non	Non
connectionStatus	Oui	Oui	Oui	Oui
dataSize	Oui	Oui	Oui	Oui
dbHash	Non	Non	Non	Non
dbStats	Oui	Oui	Oui	Oui
explain	Oui	Oui	Oui	Oui
explain: executionStats	Oui	Oui	Oui	Oui

Command	3.6	4.0	5.0	Cluster élastique
fonctionnalités	Non	Non	Non	Non
hostInfo	Oui	Oui	Oui	Oui
listCommands	Oui	Oui	Oui	Oui
Profiler	Oui	Oui	Oui	Non
serverStatus	Oui	Oui	Oui	Oui
top	Oui	Oui	Oui	Oui

Opérations d'écriture et de requête

Command	3.6	4.0	5.0	Cluster élastique
supprimer	Oui	Oui	Oui	Oui
find	Oui	Oui	Oui	Oui
trouver AndModify	Oui	Oui	Oui	Oui
obtenir LastError	Non	Non	Non	Non
getMore	Oui	Oui	Oui	Oui
obtenir PrevError	Non	Non	Non	Non
insert	Oui	Oui	Oui	Oui
parallel Collectio nScan	Non	Non	Non	Non
resetError	Non	Non	Non	Non
mise à jour	Oui	Oui	Oui	Oui

Command	3.6	4.0	5.0	Cluster élastique
Change streams	Oui	Oui	Oui	Non
GridFS	Non	Non	Non	Non
ReplaceOne	Oui	Oui	Oui	Oui

Commandes de gestion des rôles

C	3	4	5	C é
c e	O	O	O	Non
c	O	O	O	Non
l t A F a	O	O	O	Non o
F l c s n	O	O	O	Non n
r l r F s	O	O	O	Non
r l	O	O	O	Non

C	3	4	5	C é
r F				n
ıO	ıO	ıO	ıO	Non e
rO	ıO	ıO	ıO	Non

Commandes de session

Command	3.6	4.0	5.0	Cluster élastique
Annulation de la transaction	Non	Oui	Oui	Non
commitTransaction	Non	Oui	Oui	Non
Fin des sessions	Non	Non	Non	Non
killAllSessions	Non	Oui	Oui	Non
tuer AllSessions ByPattern	Non	Non	Non	Non
Kill Sessions	Non	Oui	Oui	Non
Séances de rafraîchissement	Non	Non	Non	Non
Démarrer la session	Non	Oui	Oui	Non

Gestion des utilisateurs

Command	3.6	4.0	5.0	Cluster élastique
createUser	Oui	Oui	Oui	Oui
laisser tomber AllUsers FromDatabase	Oui	Oui	Oui	Oui
dropUser	Oui	Oui	Oui	Oui
accorder un RolesTo utilisateur	Oui	Oui	Oui	Oui
révoquer un utilisateur RolesFrom	Oui	Oui	Oui	Oui
updateUser	Oui	Oui	Oui	Oui
userInfo	Oui	Oui	Oui	Oui

Commandes de partitionnement

Command	Cluster élastique
avorter ReshardCollection	Non
Ajouter un fragment	Non
ajouter une ShardTo zone	Non
équilibreur CollectionStatus	Non
BalancerStart	Non

Command	Cluster élastique
État de l'équilibreur	Non
BalancerStop	Non
vérifier ShardingIndex	Non
clair JumboFlag	Non
cleanupOrphaned	Non
nettoyage ReshardCollection	Non
commettre ReshardCollection	Non
Activer le partage	Oui
chasse d'eau RouterConfig	Non
obtenir ShardMap	Non
obtenir ShardVersion	Non
isdbgrid	Non
Listes Shards	Non
Clé médiane	Non
Déplacer Chunk	Non
Déplacer le primaire	Non
Fusionner des morceaux	Non
Affiner CollectionShard la clé	Non
Supprimer le dur	Non
supprimer ShardFrom la zone	Non

Command	Cluster élastique
Collection Reshard	Non
ensemble AllowMigrations	Non
ensemble ShardVersion	Non
Collection SHARD	Oui
État de partage	Non
split	Non
Vecteur divisé	Non
Désactiver le sharding	Non
mettre à jour ZoneKey Range	Non

Opérateurs de projection et de requête

Rubriques

- Opérateurs de grappe
- Opérateurs au niveau du bit
- Opérateur de commentaire
- Opérateurs de comparaison
- Opérateurs d'élément
- Opérateurs de requête d'évaluation
- Opérateurs logiques
- Opérateurs de projection

Opérateurs de grappe

Command	3.6	4.0	5.0	Cluster élastique
$all	Oui	Oui	Oui	Oui
$elemMatch	Oui	Oui	Oui	Oui
$size	Oui	Oui	Oui	Oui

Opérateurs au niveau du bit

Command	3.6	4.0	5.0	Cluster élastique
$bits AllSet	Oui	Oui	Oui	Oui
$bits AnySet	Oui	Oui	Oui	Oui
$bits AllClear	Oui	Oui	Oui	Oui
$bits AnyClear	Oui	Oui	Oui	Oui

Opérateur de commentaire

Command	3.6	4.0	5.0	Cluster élastique
$comment	Oui	Oui	Oui	Oui

Opérateurs de comparaison

Command	3.6	4.0	5.0	Cluster élastique
$eq	Oui	Oui	Oui	Oui
$gt	Oui	Oui	Oui	Oui

Command	3.6	4.0	5.0	Cluster élastique
$gte	Oui	Oui	Oui	Oui
$lt	Oui	Oui	Oui	Oui
$lte	Oui	Oui	Oui	Oui
$ne	Oui	Oui	Oui	Oui
$in	Oui	Oui	Oui	Oui
$nin	Oui	Oui	Oui	Oui

Opérateurs d'élément

Command	3.6	4.0	5.0	Cluster élastique
$exists	Oui	Oui	Oui	Oui
$type	Oui	Oui	Oui	Oui

Opérateurs de requête d'évaluation

Command	3.6	4.0	5.0	Cluster élastique
$expr	Non	Oui	Oui	Non
$jsonSchema	Non	Oui	Oui	Non
$mod	Oui	Oui	Oui	Oui
$regex	Oui	Oui	Oui	Oui
$text	Non	Non	Oui	Non
$where	Non	Non	Non	Non

Opérateurs logiques

Command	3.6	4.0	5.0	Cluster élastique
$or	Oui	Oui	Oui	Oui
$and	Oui	Oui	Oui	Oui
$not	Oui	Oui	Oui	Oui
$nor	Oui	Oui	Oui	Oui

Opérateurs de projection

Command	3.6	4.0	5.0	Cluster élastique
$	Oui	Oui	Oui	Oui
$elemMatch	Oui	Oui	Oui	Oui
$meta	Non	Non	Oui	Non
$slice	Oui	Oui	Oui	Oui

Opérateurs de mise à jour

Rubriques

- Opérateurs de grappe
- Opérateurs au niveau du bit
- Opérateurs de champ
- Modificateurs de mise à jour

Opérateurs de grappe

Command	3.6	4.0	5.0	Cluster élastique
$	Oui	Oui	Oui	Oui
$[]	Oui	Oui	Oui	Oui
$[<identifier>]	Oui	Oui	Oui	Oui
$ajouter ToSet	Oui	Oui	Oui	Oui
$pop	Oui	Oui	Oui	Oui
$pullAll	Oui	Oui	Oui	Oui
$pull	Oui	Oui	Oui	Oui
$push	Oui	Oui	Oui	Oui

Opérateurs au niveau du bit

Command	3.6	4.0	5.0	Cluster élastique
$bit	Oui	Oui	Oui	Oui

Opérateurs de champ

Opérateur	3.6	4.0	5.0	Cluster élastique
$inc	Oui	Oui	Oui	Oui
$mul	Oui	Oui	Oui	Oui
$rename	Oui	Oui	Oui	Oui

Opérateur	3.6	4.0	5.0	Cluster élastique
ensemble de dollars OnInsert	Oui	Oui	Oui	Oui
$set	Oui	Oui	Oui	Oui
$unset	Oui	Oui	Oui	Oui
$min	Oui	Oui	Oui	Oui
$max	Oui	Oui	Oui	Oui
$currentDate	Oui	Oui	Oui	Oui

Modificateurs de mise à jour

Opérateur	3.6	4.0	5.0	Cluster élastique
$each	Oui	Oui	Oui	Oui
$slice	Oui	Oui	Oui	Oui
$sort	Oui	Oui	Oui	Oui
$position	Oui	Oui	Oui	Oui

Géospatial

Spécificateurs de géométrie

Sélecteurs de requête	3.6	4.0	5.0	Cluster élastique
$box	Non	Non	Non	Non
$center	Non	Non	Non	Non

Sélecteurs de requête	3.6	4.0	5.0	Cluster élastique
$centerSphere	Non	Non	Non	Non
$nearSphere	Oui	Oui	Oui	Non
$geometry	Oui	Oui	Oui	Non
$maxDistance	Oui	Oui	Oui	Non
$minDistance	Oui	Oui	Oui	Non
$polygon	Non	Non	Non	Non
$uniqueDocs	Non	Non	Non	Non

Sélecteurs de requête

Command	3.6	4.0	5.0	Cluster élastique
$geoIntersects	Oui	Oui	Oui	Non
$geoWithin	Oui	Oui	Oui	Non
$near	Non	Non	Non	Non
$nearSphere	Oui	Oui	Oui	Non
$polygon	Non	Non	Non	Non
$uniqueDocs	Non	Non	Non	Non

Méthodes de curseur

Command	3.6	4.0	5.0	Cluster élastique
cursor.batchSize()	Oui	Oui	Oui	Oui
cursor.close()	Oui	Oui	Oui	Oui
cursor.isClosed()	Oui	Oui	Oui	Oui
cursor.collation()	Non	Non	Non	Non
cursor.comment()	Oui	Oui	Oui	Oui
cursor.count()	Oui	Oui	Oui	Oui
cursor.explain()	Oui	Oui	Oui	Non
cursor.forEach()	Oui	Oui	Oui	Oui
cursor.hasNext()	Oui	Oui	Oui	Oui
cursor.hint()	Oui	Oui	Oui	Oui*
cursor.isExhausted()	Oui	Oui	Oui	Non
cursor.itcount()	Oui	Oui	Oui	Non
cursor.limit()	Oui	Oui	Oui	Non
cursor.map()	Oui	Oui	Oui	Non
cursor.maxScan()	Oui	Oui	Oui	Non
cursor.maxTimeMS()	Oui	Oui	Oui	Non

Command	3.6	4.0	5.0	Cluster élastique
cursor.max()	Non	Non	Non	Non
cursor.min()	Non	Non	Non	Non
cursor.next()	Oui	Oui	Oui	Oui
curseur.no CursorTimeout ()	Non	Non	Non	Non
cursor.objs Batch () LeftIn	Oui	Oui	Oui	Non
cursor.pretty()	Oui	Oui	Oui	Non
cursor.re adConcern()	Oui	Oui	Oui	Non
cursor.readPref()	Oui	Oui	Oui	Non
cursor.re turnKey()	Non	Non	Non	Non
curseur.show RecordId ()	Non	Non	Non	Non
cursor.size()	Oui	Oui	Oui	Non
cursor.skip()	Oui	Oui	Oui	Non
cursor.sort()	Oui	Oui	Oui	Non
cursor.tailable()	Non	Non	Non	Non
cursor.toArray()	Oui	Oui	Oui	Non

* L'index `hint` est pris en charge par des expressions d'index. Par exemple, `db.foo.find().hint({x:1})`.

Opérateurs regroupement pipeline

Rubriques

Expressions accumulateur

Expression	3.6	4.0	5.0	Cluster élastique
$sum	Oui	Oui	Oui	Oui
$avg	Oui	Oui	Oui	Oui

Expression	3.6	4.0	5.0	Cluster élastique
$first	Oui	Oui	Oui	Oui
$last	Oui	Oui	Oui	Oui
$max	Oui	Oui	Oui	Oui
$min	Oui	Oui	Oui	Oui
$push	Oui	Oui	Oui	Oui
$ajouter ToSet	Oui	Oui	Oui	Oui
$std DevPop	Non	Non	Non	Non
$std DevSamp	Non	Non	Non	Non
$accumulateur	-	-	Non	Non
$count	-	-	Non	Non

Opérateurs arithmétiques

Command	3.6	4.0	5.0	Cluster élastique
$abs	Oui	Oui	Oui	Oui
$add	Oui	Oui	Oui	Oui
$ceil	Non	Oui	Oui	Oui
$divide	Oui	Oui	Oui	Oui
$exp	Non	Oui	Oui	Oui
$floor	Non	Oui	Oui	Oui
$ln	Non	Oui	Oui	Oui

Command	3.6	4.0	5.0	Cluster élastique
$log	Non	Oui	Oui	Oui
$log10	Non	Oui	Oui	Oui
$mod	Oui	Oui	Oui	Oui
$multiply	Oui	Oui	Oui	Oui
$pow	Non	Non	Non	Non
$sqrt	Non	Oui	Oui	Oui
$subtract	Oui	Oui	Oui	Oui
$trunc	Non	Non	Non	Non
$round	-	-	Non	Non

Opérateurs de grappe

Command	3.6	4.0	5.0	Cluster élastique
$array ElemAt	Oui	Oui	Oui	Oui
$array ToObject	Oui	Oui	Oui	Oui
$concatArrays	Oui	Oui	Oui	Oui
$filter	Oui	Oui	Oui	Oui
indice $ OfArray	Oui	Oui	Oui	Oui
$isArray	Oui	Oui	Oui	Oui
$objet ToArray	Oui	Oui	Oui	Oui
$range	Oui	Oui	Oui	Oui

Command	3.6	4.0	5.0	Cluster élastique
$reverseArray	Oui	Oui	Oui	Oui
$reduce	Oui	Oui	Oui	Oui
$size	Oui	Oui	Oui	Oui
$slice	Oui	Oui	Oui	Oui
$zip	Oui	Oui	Oui	Oui
$in	Oui	Oui	Oui	Oui
$first	-	-	Non	Non
$last	-	-	Non	Non

Opérateurs booléens

Command	3.6	4.0	5.0	Cluster élastique
$and	Oui	Oui	Oui	Oui
$or	Oui	Oui	Oui	Oui
$not	Oui	Oui	Oui	Oui

Opérateurs de comparaison

Command	3.6	4.0	5.0	Cluster élastique
$cmp	Oui	Oui	Oui	Oui
$eq	Oui	Oui	Oui	Oui
$gt	Oui	Oui	Oui	Oui

Command	3.6	4.0	5.0	Cluster élastique
$gte	Oui	Oui	Oui	Oui
$lt	Oui	Oui	Oui	Oui
$lte	Oui	Oui	Oui	Oui
$ne	Oui	Oui	Oui	Oui

Opérateurs d'expressions conditionnelles

Command	3.6	4.0	5.0	Cluster élastique
$cond	Oui	Oui	Oui	Oui
$ifNull	Oui	Oui	Oui	Oui
$switch	Non	Oui	Oui	Non

Opérateur de type de données

Command	3.6	4.0	5.0	Cluster élastique
$type	Oui	Oui	Oui	Oui

Opérateur de taille des données

Command	3.6	4.0	5.0	Cluster élastique
$BinarySize	-	-	Non	Non
$BSON Size	-	-	Non	Non

Opérateurs de date

Command	3.6	4.0	5.0	Cluster élastique
$dateAjouter	Non	Non	Oui	Oui
$dateSubstract	Non	Non	Oui	Oui
$ par jour OfYear	Oui	Oui	Oui	Oui
$ par jour OfMonth	Oui	Oui	Oui	Oui
$ par jour OfWeek	Oui	Oui	Oui	Oui
$year	Oui	Oui	Oui	Oui
$month	Oui	Oui	Oui	Oui
$week	Oui	Oui	Oui	Oui
$hour	Oui	Oui	Oui	Oui
$minute	Oui	Oui	Oui	Oui
$second	Oui	Oui	Oui	Oui
$millisecond	Oui	Oui	Oui	Oui
$ date ToString	Oui	Oui	Oui	Oui
Semaine $iso DayOf	Oui	Oui	Oui	Oui
$isoWeek	Oui	Oui	Oui	Oui
$ date FromParts	Non	Non	Non	Non
$ date ToParts	Non	Non	Non	Non

Command	3.6	4.0	5.0	Cluster élastique
$ date FromStrin g	Oui	Oui	Oui	Oui
$iso WeekYear	Oui	Oui	Oui	Oui
$DataTrunc	-	-	Non	Non
$DataDiff	-	-	Non	Non

Opérateur de littéral

Command	3.6	4.0	5.0	Cluster élastique
$literal	Oui	Oui	Oui	Oui

Opérateur de fusion

Command	3.6	4.0	5.0	Cluster élastique
$mergeObjects	Oui	Oui	Oui	Oui

Opérateur naturel

Command	3.6	4.0	5.0	Cluster élastique
$naturel	Oui	Oui	Oui	Oui

Opérateurs d'ensembles

Command	3.6	4.0	5.0	Cluster élastique
$setEquals	Oui	Oui	Oui	Oui

Command	3.6	4.0	5.0	Cluster élastique
$setIntersection	Oui	Oui	Oui	Oui
$setUnion	Oui	Oui	Oui	Oui
$setDifference	Non	Oui	Oui	Oui
ensemble de dollars IsSubset	Oui	Oui	Oui	Oui
$ n'importe lequel ElementTrue	Non	Oui	Oui	Oui
$ tous ElementsTrue	Non	Oui	Oui	Oui

Opérateurs d'étape

Command	3.6	4.0	5.0	Cluster élastique
$collStats	Non	Non	Non	Non
$project	Oui	Oui	Oui	Oui
$match	Oui	Oui	Oui	Oui
$redact	Oui	Oui	Oui	Oui
$limit	Oui	Oui	Oui	Oui
$skip	Oui	Oui	Oui	Oui
$unwind	Oui	Oui	Oui	Oui
$group	Oui	Oui	Oui	Oui
$sample	Oui	Oui	Oui	Oui

Command	3.6	4.0	5.0	Cluster élastique
$sort	Oui	Oui	Oui	Oui
$geoNear	Oui	Oui	Oui	Non
$lookup	Oui	Oui	Oui	Oui
$out	Oui	Oui	Oui	Non
$indexStats	Oui	Oui	Oui	Oui
$facet	Non	Non	Non	Non
$bucket	Non	Non	Non	Non
$bucketAuto	Non	Non	Non	Non
$sort ByCount	Non	Non	Non	Non
$addFields	Oui	Oui	Oui	Oui
$replaceRoot	Oui	Oui	Oui	Oui
$count	Oui	Oui	Oui	Oui
$currentOp	Oui	Oui	Oui	Oui
liste de $ LocalSessions	Non	Non	Non	Non
$listSessions	Non	Non	Non	Non
$graphLookup	Non	Non	Non	Non
$ fusion	-	-	Non	Non
$plan CacheStat s	-	-	Non	Non

Command	3.6	4.0	5.0	Cluster élastique
ensemble de dollars WindowFields	-	-	Non	Non
$ Union avec	-	-	Non	Non
$unset	-	-	Non	Non

Opérateurs de chaîne

Command	3.6	4.0	5.0	Cluster élastique
$concat	Oui	Oui	Oui	Oui
indice $ OfBytes	Oui	Oui	Oui	Oui
$indexOfCP	Oui	Oui	Oui	Oui
$ltrim	Non	Non	Non	Non
$trim	Non	Non	Non	Non
$split	Oui	Oui	Oui	Oui
$strcasecmp	Oui	Oui	Oui	Oui
$str LenBytes	Oui	Oui	Oui	Oui
$strLenCP	Oui	Oui	Oui	Oui
$substr	Oui	Oui	Oui	Oui
$substrBytes	Oui	Oui	Oui	Oui
$substrCP	Oui	Oui	Oui	Oui
$toLower	Oui	Oui	Oui	Oui

Command	3.6	4.0	5.0	Cluster élastique
$toUpper	Oui	Oui	Oui	Oui
$trim	Non	Non	Non	Non
$RegXFind	-	-	Non	Non
$regex FindAll	-	-	Non	Non
$RegexMatch	-	-	Non	Non
$RemplaceOne	-	-	Non	Non
$ Remplacer tout	-	-	Non	Non

Variables système

Command	3.6	4.0	5.0	Cluster élastique
$$CURRENT	Non	Non	Non	Non
$$DESCEND	Oui	Oui	Oui	Oui
$$KEEP	Oui	Oui	Oui	Oui
$$PRUNE	Oui	Oui	Oui	Oui
$$REMOVE	Non	Non	Non	Non
$$ROOT	Oui	Oui	Oui	Oui

Opérateur de recherche de texte

Command	3.6	4.0	5.0	Cluster élastique
$search	Non	Non	Oui	Non

Command	3.6	4.0	5.0	Cluster élastique
$meta	Non	Non	Oui	Non

Opérateurs de conversion de type

Command	3.6	4.0	5.0	Cluster élastique
$ convertir	Non	Oui	Oui	Oui
$ à Bool	Non	Oui	Oui	Oui
$ à ce jour	Non	Oui	Oui	Oui
$ en décimal	Non	Oui	Oui	Oui
$ à doubler	Non	Oui	Oui	Oui
$ en INT	Non	Oui	Oui	Oui
$ trop long	Non	Oui	Oui	Oui
$ à ObjectId	Non	Oui	Oui	Oui
$toString	Non	Oui	Oui	Oui
$isNumber	-	-	Non	Non

Opérateurs de variable

Command	3.6	4.0	5.0	Cluster élastique
$map	Oui	Oui	Oui	Oui
$let	Oui	Oui	Oui	Oui

Opérateurs divers

Command	3.6	4.0	5.0	Cluster élastique
$ rand	-	-	Non	Non
$ SampleRate	-	-	Non	Non
$GetField	-	-	Non	Non

Les types de données

Command	3.6	4.0	5.0	Cluster élastique
Double	Oui	Oui	Oui	Oui
Chaîne	Oui	Oui	Oui	Oui
Objet	Oui	Oui	Oui	Oui
Tableau	Oui	Oui	Oui	Oui
Données binaires	Oui	Oui	Oui	Oui
ObjectId	Oui	Oui	Oui	Oui
Booléen	Oui	Oui	Oui	Oui
Date	Oui	Oui	Oui	Oui
Null	Oui	Oui	Oui	Oui
Entier 32 bits (int)	Oui	Oui	Oui	Oui
Horodatage	Oui	Oui	Oui	Oui

Command	3.6	4.0	5.0	Cluster élastique
Entier 64 bits (long)	Oui	Oui	Oui	Oui
MinKey	Oui	Oui	Oui	Oui
MaxKey	Oui	Oui	Oui	Oui
Decimal128	Oui	Oui	Oui	Oui
Expression régulière	Oui	Oui	Oui	Oui
JavaScript	Non	Non	Non	Non
JavaScript(avec lunette)	Non	Non	Non	Non
Non défini	Non	Non	Non	Non
Symbol	Non	Non	Non	Non
DBPointer	Non	Non	Non	Non

Propriétés de l'index et des index

Rubriques

- Index
- Propriétés d'index

Index

Command	3.6	4.0	5.0	Cluster élastique
Index de champ unique	Oui	Oui	Oui	Oui

Command	3.6	4.0	5.0	Cluster élastique
Index composé	Oui	Oui	Oui	Oui
Index multiclés	Oui	Oui	Oui	Oui
Index de texte	Non	Non	Oui	Non
Sphère 2d	Oui	Oui	Oui	Non
Index 2d	Non	Non	Non	Non
Index haché	Non	Non	Non	Non

Propriétés d'index

Command	3.6	4.0	5.0	Cluster élastique
TTL	Oui	Oui	Oui	Oui
Unique	Oui	Oui	Oui	Oui
Partielle	Non	Non	Oui	Non
Sensible à la casse	Non	Non	Non	Non
Fragmentée	Oui	Oui	Oui	Oui
Contexte	Oui	Oui	Oui	Non

Intelligence artificielle générative Amazon DocumentDB

Amazon DocumentDB propose des fonctionnalités permettant aux modèles d'apprentissage automatique (ML) et d'intelligence artificielle générative (IA) de fonctionner avec les données stockées dans Amazon DocumentDB en temps réel. Les clients n'ont plus à perdre de temps à gérer une infrastructure distincte, à écrire du code pour se connecter à un autre service et à dupliquer les données de leur base de données principale.

Pour plus d'informations sur l'intelligence artificielle et sur la manière dont AWS vous pouvez répondre à vos besoins en matière d'IA, consultez cet article « Qu'est-ce que c'est ».

Rubriques

- Apprentissage automatique sans code avec Amazon Canvas SageMaker
- Recherche vectorielle pour Amazon DocumentDB

Apprentissage automatique sans code avec Amazon Canvas SageMaker

Amazon SageMaker Canvas vous permet de créer vos propres modèles d'IA/ML sans avoir à écrire une seule ligne de code. Vous pouvez créer des modèles de machine learning pour des cas d'utilisation courants tels que la régression et les prévisions, et vous pouvez accéder à des modèles de base (FM) et les évaluer depuis Amazon Bedrock. Vous pouvez également accéder aux FM publiques d'Amazon SageMaker JumpStart pour la génération de contenu, l'extraction de texte et la synthèse de texte afin de prendre en charge les solutions d'IA génératives.

Comment créer des modèles ML sans code avec Canvas SageMaker

Amazon DocumentDB s'intègre désormais à Amazon SageMaker Canvas pour permettre l'apprentissage automatique (ML) sans code avec les données stockées dans Amazon DocumentDB. Vous pouvez désormais créer des modèles de machine learning pour les besoins de régression et de prévision et utiliser des modèles de base pour la synthèse et la génération de contenu à l'aide de données stockées dans Amazon DocumentDB sans écrire une seule ligne de code.

SageMaker Canvas fournit une interface visuelle qui permet aux clients d'Amazon DocumentDB de générer des prédictions sans avoir besoin d'expertise en intelligence artificielle ou en machine learning ou d'écrire une seule ligne de code. Les clients peuvent désormais lancer l'espace de travail

SageMaker Canvas à partir des données Amazon DocumentDB AWS Management Console, les importer et les joindre à des fins de préparation des données et de formation des modèles. Les données d'Amazon DocumentDB peuvent désormais être utilisées dans SageMaker Canvas pour créer et améliorer des modèles destinés à prévoir le taux de désabonnement des clients, à détecter les fraudes, à prévoir les défaillances de maintenance, à prévoir les indicateurs commerciaux et à générer du contenu. Les clients peuvent désormais publier et partager des informations basées sur le ML entre les équipes grâce à l'intégration native de SageMaker Canvas avec Amazon. QuickSight Les pipelines d'ingestion de données dans SageMaker Canvas s'exécutent par défaut sur les instances secondaires d'Amazon DocumentDB, ce qui garantit que les performances des applications et des charges de travail d'ingestion de SageMaker Canvas ne sont pas entravées.

Les clients Amazon DocumentDB peuvent commencer à utiliser SageMaker Canvas en accédant à la nouvelle page de la console Amazon DocumentDB No-Code ML et en se connectant à des espaces de travail Canvas nouveaux ou disponibles. SageMaker

Configuration du SageMaker domaine et du profil utilisateur

Vous pouvez vous connecter aux clusters Amazon DocumentDB à partir de SageMaker domaines exécutés en mode VPC uniquement. En lançant un SageMaker domaine dans votre VPC, vous pouvez contrôler le flux de données depuis vos environnements SageMaker Studio et Canvas. Cela vous permet de restreindre l'accès à Internet, de surveiller et d'inspecter le trafic à l'aide de fonctionnalités AWS réseau et de sécurité standard, et de vous connecter à d'autres AWS ressources via des points de terminaison VPC. Reportez-vous à Amazon SageMaker Canvas Getting started and Configure Amazon SageMaker Canvas dans un VPC sans accès Internet, qui se trouve dans le manuel Amazon SageMaker Developer Guide pour créer votre SageMaker domaine afin de vous connecter à votre cluster Amazon DocumentDB.

Configuration des autorisations d'accès IAM pour Amazon SageMaker DocumentDB et Canvas

Un utilisateur Amazon DocumentDB `AmazonDocDBConsoleFullAccess` attaché au rôle et à l'identité qui lui sont associés peut accéder au. AWS Management Console Ajoutez les actions suivantes au rôle ou à l'identité susmentionnés pour permettre l'accès à l'apprentissage automatique sans code avec Amazon SageMaker Canvas.

```
"sagemaker:CreatePresignedDomainUrl",
"sagemaker:DescribeDomain",
"sagemaker:ListDomains",
```

```
"sagemaker:ListUserProfiles"
```

Création d'utilisateurs et de rôles de base de données pour SageMaker Canvas

Vous pouvez restreindre l'accès aux actions que les utilisateurs peuvent effectuer sur les bases de données à l'aide du contrôle d'accès basé sur les rôles (RBAC) dans Amazon DocumentDB. Le RBAC fonctionne en accordant un ou plusieurs rôles à un utilisateur. Ces rôles déterminent les opérations qu'un utilisateur peut effectuer sur les ressources de base de données.

En tant qu'utilisateur de Canvas, vous vous connectez à une base de données Amazon DocumentDB avec un nom d'utilisateur et un mot de passe. Vous pouvez créer un utilisateur/un rôle de base de données pour un utilisateur de Canvas disposant d'un accès en lecture aux bases de données spécifiques à l'aide de la fonctionnalité RBAC d'Amazon DocumentDBB.

Par exemple, utilisez l'`createUser`opération :

```
db.createUser({
user: "canvas_user",
pwd: "<insert-password>",
roles: [{role: "read", db: "sample-database-1"}]
})
```

Cela crée un `canvas_user` qui dispose d'autorisations de lecture sur la `sample-database-1` base de données. Vos analystes Canvas peuvent utiliser ces informations d'identification pour accéder aux données de votre cluster Amazon DocumentDB. Reportez-vous Accès à la base de données à l'aide du contrôle d'accès basé sur les rôles à pour en savoir plus.

Régions disponibles

L'intégration sans code est disponible dans les régions où Amazon DocumentDB et SageMaker Amazon Canvas sont pris en charge. Les régions incluent :

- us-east-1 (Virginie du Nord)
- us-east-2 (Ohio)
- us-west-2 (Oregon)
- ap-northeast-1 (Tokyo)
- ap-northeast-2 (Séoul)

- ap-south-1 (Bombay)
- ap-southeast-1 (Singapour)
- ap-southeast-2 (Sydney)
- eu-central-1 (Francfort)
- eu-west-1 (Irlande)

Reportez-vous à Amazon SageMaker Canvas dans le guide du SageMaker développeur Amazon pour connaître les dernières régions disponibles.

Recherche vectorielle pour Amazon DocumentDB

La recherche vectorielle est une méthode utilisée en apprentissage automatique pour trouver des points de données similaires à un point de données donné en comparant leurs représentations vectorielles à l'aide de métriques de distance ou de similarité. Plus les deux vecteurs sont proches de l'espace vectoriel, plus les éléments sous-jacents sont considérés comme similaires. Cette technique permet de saisir le sens sémantique des données. Cette approche est utile dans diverses applications, telles que les systèmes de recommandation, le traitement du langage naturel et la reconnaissance d'images.

La recherche vectorielle pour Amazon DocumentDB associe la flexibilité et la riche capacité d'interrogation d'une base de données de documents basée sur JSON à la puissance de la recherche vectorielle. Si vous souhaitez utiliser vos données Amazon DocumentDB existantes ou une structure de données documentaire flexible pour créer des cas d'utilisation de l'apprentissage automatique et de l'IA générative, tels que l'expérience de recherche sémantique, la recommandation de produits, la personnalisation, les chatbots, la détection de fraudes et la détection d'anomalies, la recherche vectorielle pour Amazon DocumentDB est le choix idéal pour vous. La recherche vectorielle est disponible sur les clusters basés sur des instances Amazon DocumentDB 5.0.

Rubriques

Insertion de vecteurs

Pour insérer des vecteurs dans votre base de données Amazon DocumentDB, vous pouvez utiliser les méthodes d'insertion existantes :

Exemple

Dans l'exemple suivant, une collection de cinq documents dans une base de données de test est créée. Chaque document comprend deux champs : le nom du produit et son incorporation vectorielle correspondante.

```
db.collection.insertMany([
  {"product_name": "Product A", "vectorEmbedding": [0.2, 0.5, 0.8]},
  {"product_name": "Product B", "vectorEmbedding": [0.7, 0.3, 0.9]},
  {"product_name": "Product C", "vectorEmbedding": [0.1, 0.2, 0.5]},
  {"product_name": "Product D", "vectorEmbedding": [0.9, 0.6, 0.4]},
  {"product_name": "Product E", "vectorEmbedding": [0.4, 0.7, 0.2]}
]);
```

Création d'un index vectoriel

Amazon DocumentDB prend en charge à la fois les méthodes d'indexation Hierarchical Navigable Small World (HNSW) et les méthodes d'indexation de fichiers inversés avec compression plate (IVFFlat). Un index IVFFlat sépare les vecteurs en listes et recherche ensuite un sous-ensemble sélectionné de ces listes les plus proches du vecteur de requête. D'autre part, un indice HNSW organise les données vectorielles dans un graphique multicouche. Bien que les temps de construction de HNSW soient plus lents que ceux d'IVFFlat, il offre de meilleures performances de requête et un meilleur rappel. Contrairement à IVFFlat, HNSW ne comporte aucune étape d'apprentissage, ce qui permet de générer l'index sans aucun chargement de données initial. Dans la majorité des cas d'utilisation, nous recommandons d'utiliser le type d'index HNSW pour la recherche vectorielle.

Si vous ne créez pas d'index vectoriel, Amazon DocumentDB effectue une recherche avec le voisin le plus proche, garantissant ainsi un rappel parfait. Cependant, dans les scénarios de production, la rapidité est cruciale. Nous vous recommandons d'utiliser des index vectoriels, qui peuvent échanger un certain rappel contre une amélioration de la vitesse. Il est important de noter que l'ajout d'un index vectoriel peut entraîner des résultats de requête différents.

Modèles

Vous pouvez utiliser les `runCommand` modèles suivants `createIndex` pour créer un index vectoriel sur un champ vectoriel :

Using createIndex

Dans certains pilotes, tels que Mongosh et Java, l'utilisation `vectorOptions` des paramètres `createIndex` peut entraîner une erreur. Dans de tels cas, nous vous recommandons d'utiliser `runCommand` :

```
db.collection.createIndex(
  { "<vectorField>": "vector" },
  { "name": "<indexName>",
    "vectorOptions": {
      "type": " <hnsw> | <ivfflat> ",
      "dimensions": <number_of_dimensions>,
      "similarity": " <euclidean> | <cosine> | <dotProduct> ",
      "lists": <number_of_lists> [applicable for IVFFlat],
      "m": <max number of connections> [applicable for HNSW],
      "efConstruction": <size of the dynamic list for index build> [applicable for
 HNSW]
    }
  }
);
```

Using runCommand

Dans certains pilotes, tels que Mongosh et Java, l'utilisation `vectorOptions` des paramètres `createIndex` peut entraîner une erreur. Dans de tels cas, nous vous recommandons d'utiliser `runCommand` :

```
db.runCommand(
  { "createIndexes": "<collection>",
  "indexes": [{
      key: { "<vectorField>": "vector" },
      vectorOptions: {
          type: " <hnsw> | <ivfflat> ",
          dimensions: <number of dimensions>,
          similarity: " <euclidean> | <cosine> | <dotProduct> ",
          lists: <number_of_lists> [applicable for IVFFlat],
          m: <max number of connections> [applicable for HNSW],
          efConstruction: <size of the dynamic list for index build> [applicable for
 HNSW]
```

```
            },
        name: "myIndex"
        }]
    }
);
```

Paramètre	Exigence	Type de données	Description	Valeur (s)
name	facultatif	chaîne	Spécifie le nom de l'index.	Alphanumérique
type	facultatif		Spécifie le type d'index.	Supporté : hnsw ou ivfflat Par défaut : HNSW (patch moteur 3.0.4574 et versions ultérieures)
dimensions	obligatoire	entier	Spécifie le nombre de dimensions des données vectorielles.	Maximum de 2 000 dimensions.
similarity	obligatoire	chaîne	Spécifie la métrique de distance utilisée pour le calcul de similarité.	• **euclidean** • **cosine** • **dotProduct**
lists	requis pour IVFFlat	entier	Spécifie le nombre de clusters utilisés par l'index IVFFlat pour	Minimum : 1 Maximum : reportez-vous au tableau

Paramètre	Exigence	Type de données	Description	Valeur (s)
			regrouper les données vectorielles. Le paramètre recommandé est le nombre de documents /1000 pour un maximum de 1 million de documents et `sqrt(# of documents)` pour plus d'un million de documents.	des listes par type d'instance Caractéristiques et limites ci-dessous.
`m`	facultatif	entier	Spécifie le nombre maximum de connexions pour un index HNSW	Par défaut: 16 Gamme [2, 100]

Paramètre	Exigence	Type de données	Description	Valeur (s)
efConstruction	facultatif	entier	Spécifie la taille de la liste dynamique de candidats pour la construction du graphique pour l'indice HNSW. `efConstruction` doit être supérieur ou égal à (2 * m)	Par défaut: 64 Gamme [4, 1000]

Il est important que vous définissiez correctement la valeur des sous-paramètres tels que `lists` pour IVFFlat `m` et `efConstruction` pour HNSW, car cela affectera la précision/le rappel, le temps de création et les performances de votre recherche. Une valeur de liste plus élevée augmente la vitesse de la requête car elle réduit le nombre de vecteurs dans chaque liste, ce qui réduit la taille des régions. Cependant, une taille de région plus petite peut entraîner un plus grand nombre d'erreurs de rappel, ce qui se traduit par une baisse de la précision. Pour HNSW, l'augmentation de la valeur `m` et de la précision de l'index `efConstruction` augmente, tout en augmentant le temps et la taille de l'index. Voir les exemples suivants :

Exemples

HNSW

```
db.collection.createIndex(
  { "vectorEmbedding": "vector" },
  { "name": "myIndex",
    "vectorOptions": {
      "type": "hnsw",
      "dimensions": 3,
      "similarity": "euclidean",
      "m": 16,
      "efConstruction": 64
    }
```

```
  }
);
```

IVFFlat

```
db.collection.createIndex(
  { "vectorEmbedding": "vector" },
  { "name": "myIndex",
    "vectorOptions": {
      "type": "ivfflat",
      "dimensions": 3,
      "similarity": "euclidean",
      "lists":1
    }
  }
)
```

Obtenir une définition d'index

Vous pouvez consulter les détails de vos index, y compris les index vectoriels, à l'aide de la `getIndexes` commande suivante :

Exemple

```
db.collection.getIndexes()
```

Exemple de sortie

```
[
 {
  "v" : 4,
  "key" : {
   "_id" : 1
  },
  "name" : "_id_",
  "ns" : "test.collection"
 },
 {
  "v" : 4,
  "key" : {
   "vectorEmbedding" : "vector"
```

```
  },
  "name" : "myIndex",
  "vectorOptions" : {
   "type" : "ivfflat",
   "dimensions" : 3,
   "similarity" : "euclidean",
   "lists" : 1
  },
  "ns" : "test.collection"
 }
]
```

Vecteurs d'interrogation

Modèle de requête vectorielle

Utilisez le modèle suivant pour interroger un vecteur :

```
db.collection.aggregate([
  {
    $search: {
      "vectorSearch": {
        "vector": <query vector>,
        "path": "<vectorField>",
        "similarity": "<distance metric>",
        "k": <number of results>,
        "probes":<number of probes> [applicable for IVFFlat],
        "efSearch":<size of the dynamic list during search> [applicable for HNSW]
      }
    }
  }
]);
```

Paramètre	Exigence	Type	Description	Valeur (s)
vectorSearch	obligatoire	opérateur	Utilisée dans la commande $search pour interroger les vecteurs.	

Paramètre	Exigence	Type	Description	Valeur (s)
`vector`	obligatoire	array	Indique le vecteur de requête qui sera utilisé pour trouver des vecteurs similaires.	
`path`	obligatoire	chaîne	Définit le nom du champ vectoriel.	
`k`	obligatoire	entier	Spécifie le nombre de résultats renvoyés par la recherche.	
`similarity`	obligatoire	chaîne	Spécifie la métrique de distance utilisée pour le calcul de similarité.	• **`euclidean`** • **`cosine`** • **`dotProduct`**

Paramètre	Exigence	Type	Description	Valeur (s)
`probes`	facultatif	entier	Le nombre de clusters que la recherche vectorielle doit inspecter . Une valeur plus élevée permet un meilleur rappel au détriment de la rapidité. Il peut être défini sur le nombre de listes pour la recherche du voisin le plus proche exact (auquel cas le planificateur n'utilisera pas l'index). Le paramètre recommandé pour commencer le réglage fin est`sqrt(# of lists)`.	Valeur par défaut : 1

Paramètre	Exigence	Type	Description	Valeur (s)
`efSearch`	facultatif	entier	Spécifie la taille de la liste dynamique de candidats utilisée par l'index HNSW lors de la recherche . Une valeur plus élevée de `efSearch` permet un meilleur rappel au détriment de la rapidité.	Valeur par défaut : 40 Gamme [1, 1000]

Il est important de régler avec précision la valeur de `efSearch` (HNSW) ou `probes` (IVFlat) pour obtenir les performances et la précision souhaitées. Consultez les exemples d'opérations suivants :

HNSW

```
db.collection.aggregate([
  {
    $search: {
      "vectorSearch": {
        "vector": [0.2, 0.5, 0.8],
        "path": "vectorEmbedding",
        "similarity": "euclidean",
        "k": 2,
        "efSearch": 40
      }
    }
  }
]);
```

IVFFlat

```
db.collection.aggregate([
```

```
  {
    $search: {
      "vectorSearch": {
        "vector": [0.2, 0.5, 0.8],
        "path": "vectorEmbedding",
        "similarity": "euclidean",
        "k": 2,
        "probes": 1
      }
    }
  }
]);
```

Exemple de sortie

Le résultat de cette opération ressemble à ce qui suit :

```
{ "_id" : ObjectId("653d835ff96bee02cad7323c"), "product_name" : "Product A",
 "vectorEmbedding" : [ 0.2, 0.5, 0.8 ] }
{ "_id" : ObjectId("653d835ff96bee02cad7323e"), "product_name" : "Product C",
 "vectorEmbedding" : [ 0.1, 0.2, 0.5 ] }
```

Caractéristiques et limites

Compatibilité des versions

- La recherche vectorielle pour Amazon DocumentDB n'est disponible que sur les clusters basés sur des instances Amazon DocumentDB 5.0.

Vecteurs

- Amazon DocumentDB peut indexer des vecteurs de 2 000 dimensions maximum. Cependant, il est possible de stocker jusqu'à 16 000 dimensions sans index.

Index

- Pour la création d'un index IVFFlat, le paramètre recommandé pour le paramètre des listes est le nombre de documents/1000 pour un maximum de 1 million de documents et `sqrt(# of documents)` pour plus d'un million de documents. En raison d'une limite de mémoire de travail,

Amazon DocumentDB prend en charge une certaine valeur maximale du paramètre des listes en fonction du nombre de dimensions. À titre de référence, le tableau suivant fournit les valeurs maximales du paramètre des listes pour les vecteurs de 500, 1 000 et 2 000 dimensions :

Type d'instance	Listes de 500 dimensions	Listes de 1000 dimensions	Listes de 2000 dimensions
t3.med	372	257	150
r5.l	915	741	511
r5.xl	1 393	1 196	901
r5.2xl	5 460	5 230	4 788
r5,4xl	7 842	7 599	7 138
r5,8xl	11 220	10 974	10 498
r5,12xl	13 774	13 526	13 044
r5,16xl	15 943	15 694	15 208
r5,24xl	19 585	19 335	18 845

- Aucune autre option d'index telle que `compound` `sparse` ou n'est prise en charge par `partial` les index vectoriels.
- La création d'index parallèle n'est pas prise en charge pour l'indice HNSW. Il n'est pris en charge que pour l'index IVFFlat.

Requête vectorielle

- Pour les requêtes de recherche vectorielle, il est important d'affiner les paramètres tels que `probes` ou `efSearch` pour obtenir des résultats optimaux. Plus la valeur `probes` ou le `efSearch` paramètre est élevée, plus le rappel est élevé et plus la vitesse est faible. Le réglage recommandé pour commencer à affiner le paramètre des sondes est`sqrt(# of lists)`.

Bonnes pratiques

Découvrez les meilleures pratiques relatives à l'utilisation de la recherche vectorielle dans Amazon DocumentDB. Cette section est mise à jour en continu à mesure que de nouvelles bonnes pratiques sont identifiées.

- La création d'un index de fichier inversé avec compression plate (IVFFlat) implique le regroupement et l'organisation des points de données en fonction des similitudes. Par conséquent, pour qu'un index soit plus efficace, nous vous recommandons de charger au moins certaines données avant de créer l'index.
- Pour les requêtes de recherche vectorielle, il est important d'affiner les paramètres, par exemple `probes` ou `efSearch` pour obtenir des résultats optimaux. Plus la valeur du `efSearch` paramètre `probes` or est élevée, plus le rappel est élevé et plus la vitesse est faible. Le réglage recommandé pour commencer à affiner le `probes` paramètre est`sqrt(lists)`.

Ressources

- Recherche vectorielle : quels sont les nouveaux articles de blog ?
- Exemple de code de recherche sémantique
- Exemples de code de recherche vectorielle Amazon DocumentDB

Migration vers Amazon DocumentDB

Amazon DocumentDB (compatible avec MongoDB) est un service de base de données entièrement géré compatible avec l'API MongoDB. Vous pouvez migrer vos données vers Amazon DocumentDB à partir de bases de données MongoDB exécutées sur site ou sur Amazon Elastic Compute Cloud (Amazon EC2) en suivant le processus détaillé dans cette section.

Rubriques

Mise à niveau de votre cluster Amazon DocumentDB à l'aide de AWS Database Migration Service

> **Important**
>
> Amazon DocumentDB ne suit pas les mêmes cycles de support que MongoDB et le calendrier de MongoDB ne s'applique pas à Amazon end-of-life DocumentDB. Aucun plan n'est actuellement prévu end-of-life pour Amazon DocumentDB 3.6, et vos pilotes, applications et outils MongoDB 3.6 existants continueront de fonctionner avec Amazon DocumentDB.

Vous pouvez mettre à niveau votre cluster Amazon DocumentDB vers une version supérieure avec un temps d'arrêt minimal. AWS DMS AWS DMS est un service entièrement géré qui facilite la migration des anciennes versions d'Amazon DocumentDB, des bases de données relationnelles et des bases de données non relationnelles vers votre cluster Amazon DocumentDB cible.

Rubriques

Étape 1 : activer Change Streams

Pour effectuer une migration avec un temps d'arrêt minimal, AWS DMS il faut accéder aux flux de modifications du cluster. Les flux de modifications Amazon DocumentDB fournissent une séquence chronologique d'événements de mise à jour qui se produisent dans les collections et les bases de données de votre cluster. La lecture depuis le flux de modifications permet d' AWS DMS effectuer la capture des données de modification (CDC) et d'appliquer des mises à jour incrémentielles au cluster Amazon DocumentDB cible.

Pour activer les flux de modification pour toutes les collections d'une base de données spécifique, authentifiez-vous auprès de votre cluster Amazon DocumentDB à l'aide du shell mongo et exécutez les commandes suivantes :

```
db.adminCommand({modifyChangeStreams: 1,
    database: "db_name",
    collection: "",
    enable: true});
```

Étape 2 : Modifier la durée de conservation des flux de modifications

Modifiez ensuite la période de rétention du flux de modifications en fonction de la durée pendant laquelle vous souhaitez conserver les événements de changement dans le flux de modifications. Par exemple, si vous prévoyez que la migration de votre cluster Amazon DocumentDB prendra 12 heures, vous devez définir la durée de rétention du flux de modifications sur une valeur supérieure à 12 heures. AWS DMS La période de rétention par défaut pour votre cluster Amazon DocumentDB est de trois heures. Vous pouvez modifier la durée de conservation du journal des flux de modifications pour votre cluster Amazon DocumentDB pour qu'elle soit comprise entre une heure et sept jours en utilisant le AWS Management Console ou le. AWS CLI Pour plus de détails, reportez-vous à la section Modification de la durée de conservation du journal du flux de modifications.

Étape 3 : migrer vos index

Créez les mêmes index sur votre cluster Amazon DocumentDB cible que ceux que vous avez sur votre cluster Amazon DocumentDB source. Bien qu'il AWS DMS gère la migration des données, il ne migre pas les index. Pour migrer les index, utilisez l'outil d'indexation Amazon DocumentDB pour exporter les index depuis le cluster Amazon DocumentDB source. Vous pouvez obtenir l'outil en créant un clone du GitHub référentiel d'outils Amazon DocumentDB et en suivant les instructions fournies dans. `README.md` Vous pouvez exécuter l'outil à partir d'une instance Amazon EC2 ou d'un AWS Cloud9 environnement exécuté dans le même Amazon VPC que votre cluster Amazon DocumentDB.

Dans les exemples suivants, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations.

Le code suivant extrait les index de votre cluster Amazon DocumentDB source :

```
python migrationtools/documentdb_index_tool.py --dump-indexes
--uri mongodb://sample-user:user-password@sample-source-cluster.node.us-
east 1.docdb.amazonaws.com:27017/?tls=true&tlsCAFile=global-
bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false'
--dir ~/index.js/

2020-02-11 21:51:23,245: Successfully authenticated to database: admin2020-02-11
 21:46:50,432: Successfully connected to instance docdb-40-xx.cluster-xxxxxxxx.us-
east-1.docdb.amazonaws.com:27017
2020-02-11 21:46:50,432: Retrieving indexes from server...2020-02-11 21:46:50,440:
 Completed writing index metadata to local folder: /home/ec2-user/index.js/
```

Une fois vos index exportés avec succès, restaurez ces index dans votre cluster Amazon DocumentDB cible. Pour restaurer les index que vous avez exportés à l'étape précédente, utilisez l'outil d'indexation Amazon DocumentDB. La commande suivante restaure les index de votre cluster Amazon DocumentDB cible à partir du répertoire spécifié.

```
python migrationtools/documentdb_index_tool.py --restore-indexes
--uri mongodb://sample-user:user-password@sample-destination-
cluster.node.us-east 1.docdb.amazonaws.com:27017/?tls=true&tlsCAFile=global-
bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false'
--dir ~/index.js/

2020-02-11 21:51:23,245: Successfully authenticated to database: admin2020-02-11
 21:51:23,245: Successfully connected to instance docdb-50-xx.cluster-xxxxxxxx.us-
east-1.docdb.amazonaws.com:27017
2020-02-11 21:51:23,264: testdb.coll: added index: _id
```

Pour vérifier que vous avez correctement restauré les index, connectez-vous à votre cluster Amazon DocumentDB cible à l'aide du shell mongo et listez les index d'une collection donnée. Consultez le code suivant :

```
mongo --ssl
--host docdb-xx-xx.cluster-xxxxxxxx.us-east-1.docdb.amazonaws.com:27017
--sslCAFile rds-ca-2019-root.pem --username documentdb --password documentdb

db.coll.getIndexes()
```

Étape 4 : Création d'une instance AWS DMS de réplication

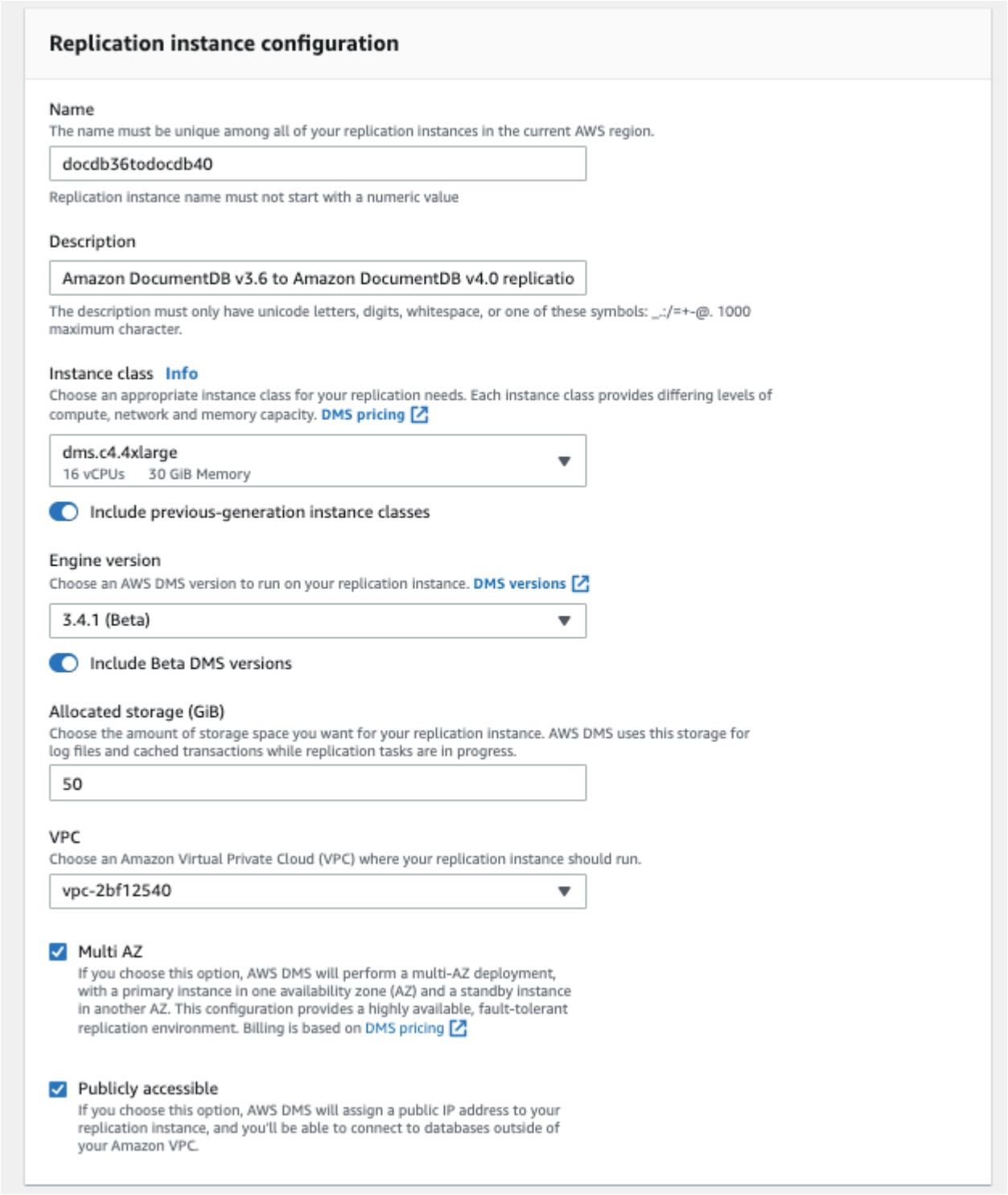
Une instance de AWS DMS réplication connecte et lit les données de votre cluster Amazon DocumentDB source et les écrit dans votre cluster Amazon DocumentDB cible. L'instance de AWS DMS réplication peut effectuer à la fois des opérations de chargement en masse et des opérations CDC. La plupart de ces traitements se font en mémoire. Toutefois, les opérations de grande envergure peuvent nécessiter une certaine mise en mémoire tampon sur le disque. Les transactions mises en cache et les fichiers journaux sont également écrits sur le disque. Une fois les données migrées, l'instance de réplication diffuse également tous les événements de modification pour s'assurer que la source et la cible sont synchronisées.

Pour créer une instance AWS DMS de réplication :

1. Ouvrez la AWS DMS console.

2. Dans le volet de navigation, sélectionnez Instances de réplication.
3. Choisissez Create replication instance (Créer une instance de réplication) et indiquez les informations suivantes :

   - Dans Nom, entrez le nom de votre choix. Par exemple, `docdb36todocdb40`.
   - Dans Description, entrez la description de votre choix. Pour listitem, instance de réplication Amazon DocumentDB 3.6 vers Amazon DocumentDB 4.0.
   - Pour la classe d'instance, choisissez la taille en fonction de vos besoins.
   - Pour la version du moteur, choisissez `3.4.1.`
   - Pour Amazon VPC, choisissez l'Amazon VPC qui héberge vos clusters Amazon DocumentDB source et cible.
   - Pour le stockage alloué (GiB), utilisez la valeur par défaut de 50 GiB. Si votre charge de travail est élevée en termes de débit d'écriture, augmentez cette valeur pour l'adapter à votre charge de travail.
   - Pour le mode Multi-AZ, choisissez Oui si vous avez besoin d'une haute disponibilité et d'une assistance en cas de basculement.
   - Pour Accessible publiquement, activez cette option.



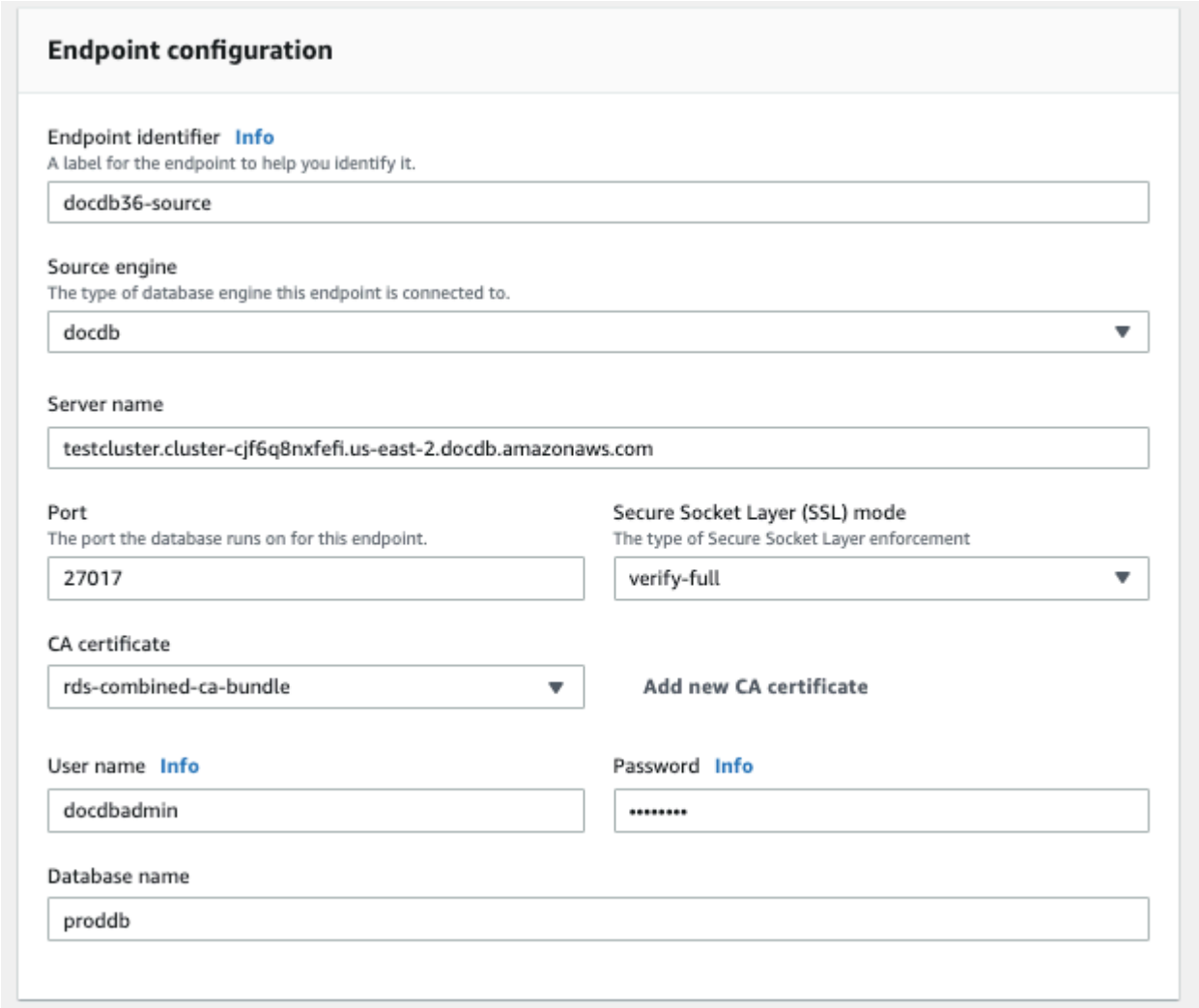
4. Choisissez Créer instance de réplication.

Étape 5 : Création d'un point de terminaison AWS DMS source

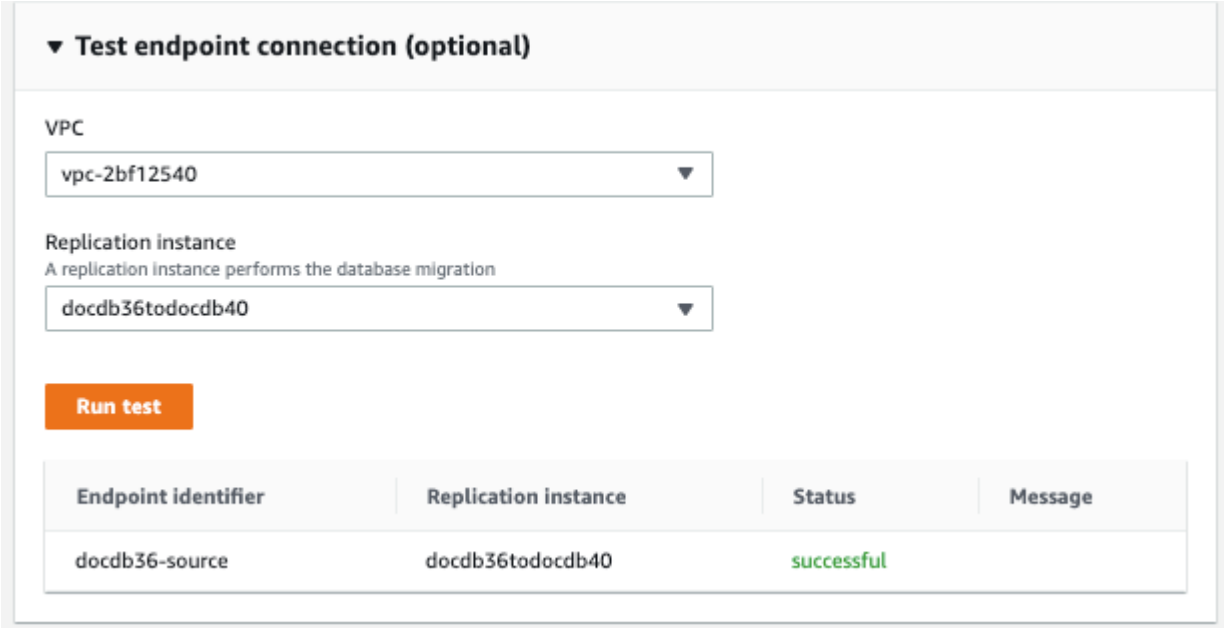
Le point de terminaison source est utilisé pour le cluster Amazon DocumentDB source.

Pour créer un point de terminaison source

1. Ouvrez la AWS DMS console.
2. Dans le panneau de navigation, choisissez Points de terminaison.
3. Choisissez `Create endpoint` et saisissez les informations suivantes :
   - Pour Type de point de terminaison, choisissez Source.
   - >Pour l'identifiant du point de terminaison, entrez un nom facile à retenir, par exemple`docdb-source`.
   - Pour le moteur source, sélectionnez`docdb`.
   - Dans Nom du serveur, entrez le nom DNS de votre cluster Amazon DocumentDB source.
   - Pour Port, entrez le numéro de port de votre cluster Amazon DocumentDB source.
   - Pour le mode SSL, choisissez`verify-full`.
   - Pour le certificat CA, choisissez Ajouter un nouveau certificat CA. Téléchargez le nouveau certificat CA (nouveau certificat pour créer un bundle de connexions TLS. Dans le champ Identifiant du certificat, entrez `rds-combined-ca-bundle`. Dans Import certificate file (Importer un fichier de certificat), choisissez Choose file (Choisir un fichier) et accédez au fichier `.pem` que vous avez téléchargé précédemment. Sélectionnez et ouvrez le fichier. Choisissez Importer un certificat, puis `rds-combined-ca-bundle` choisissez dans le menu déroulant Choisir un certificat
   - Dans Nom d'utilisateur, entrez le nom d'utilisateur principal de votre cluster Amazon DocumentDB source.
   - Pour Mot de passe, entrez le mot de passe principal de votre cluster Amazon DocumentDB source.
   - Dans Nom de la base de données, entrez le nom de la base de données que vous souhaitez mettre à niveau.



4. Testez votre connexion pour vérifier qu'elle a été correctement configurée.



5. Choisissez Créer un point de terminaison.

> **Note**
>
> AWS DMS ne peut migrer qu'une seule base de données à la fois.

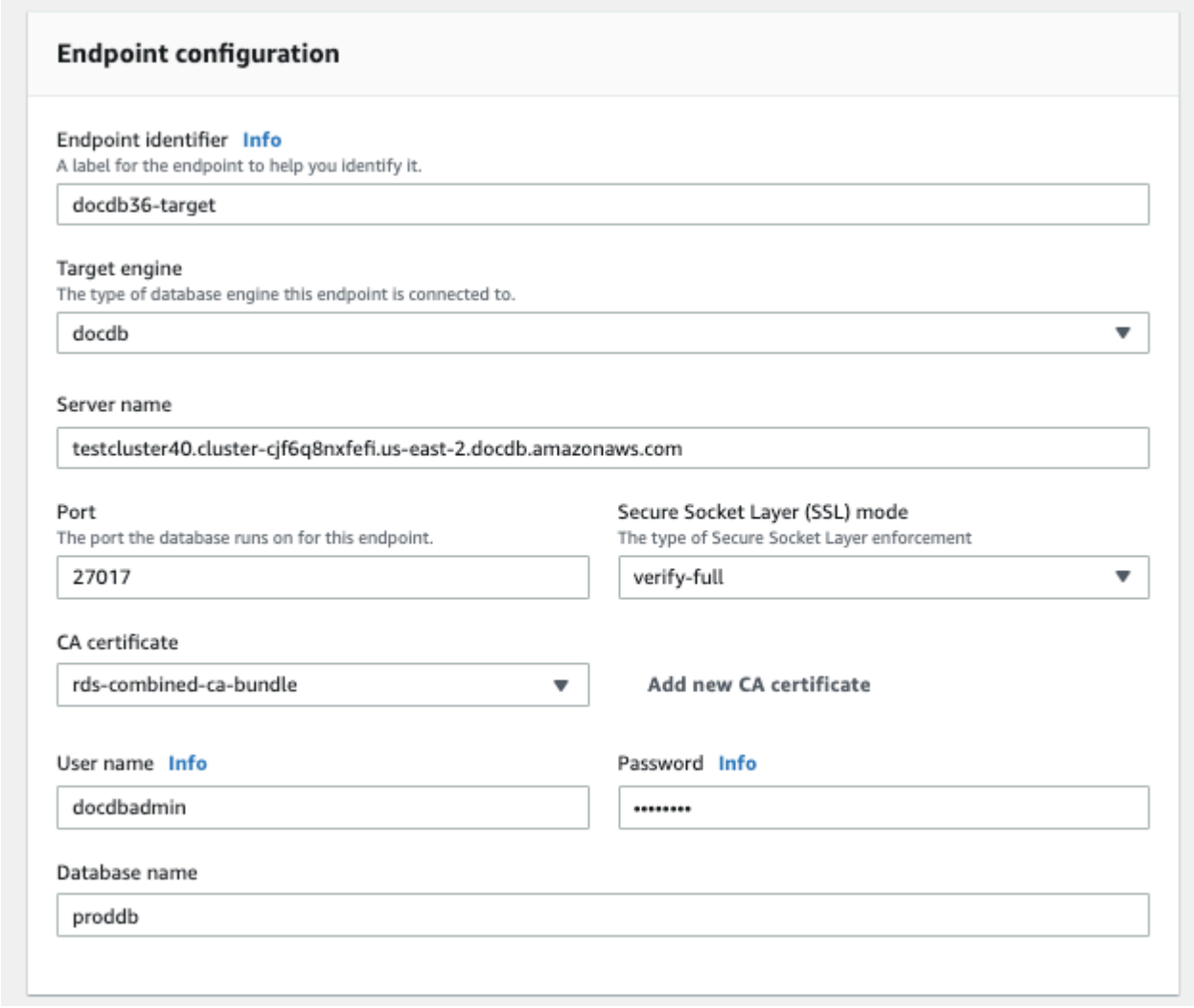
Étape 6 : Création d'un point de terminaison AWS DMS cible

Le point de terminaison cible correspond à votre cluster Amazon DocumentDB cible.

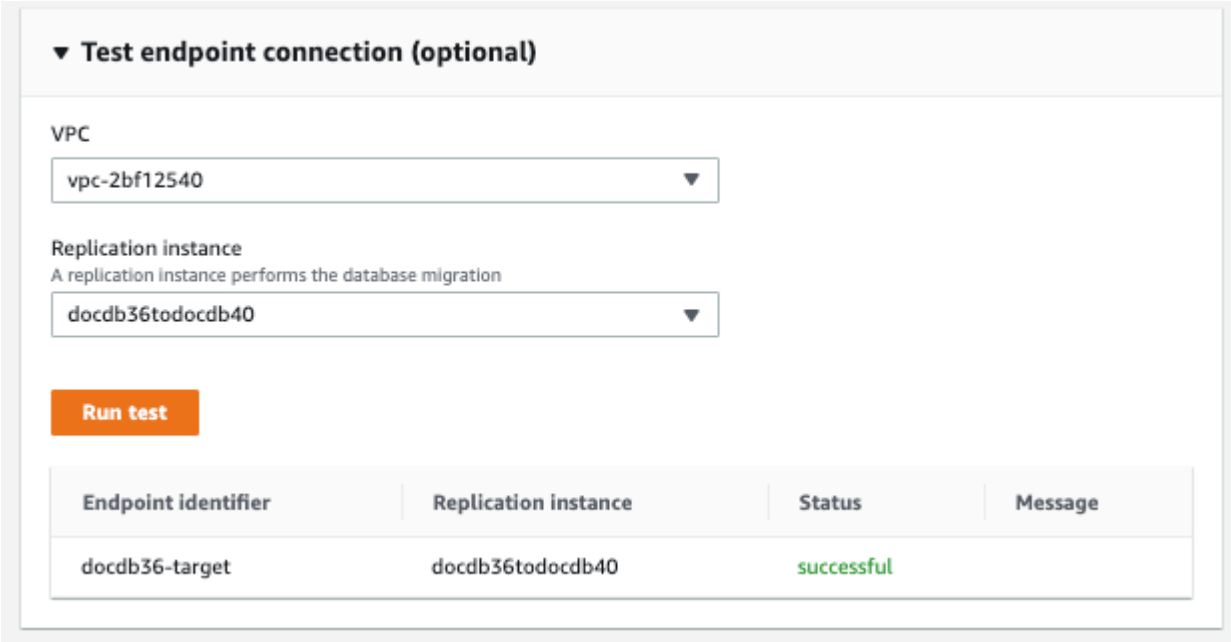
Pour créer un point de terminaison cible :

1. Ouvrez la AWS DMS console.
2. Dans le panneau de navigation, choisissez Points de terminaison.
3. Choisissez Créer un point de terminaison et entrez les informations suivantes :
   - Pour Type de point de terminaison, choisissez Cible.
   - Pour ID du point de terminaison, entrez un nom facile à mémoriser, par exemple `docdb-target`.
   - Pour le moteur source, sélectionnez`docdb`.
   - Dans Nom du serveur, entrez le nom DNS de votre cluster Amazon DocumentDB cible.

- Pour Port, entrez le numéro de port de votre cluster Amazon DocumentDB cible.
- Pour le mode SSL, choisissez`verify-full`.
- Pour le certificat CA, choisissez le `rds-combined-ca-bundle` certificat existant dans le menu déroulant Choisir un certificat.
- Dans Nom d'utilisateur, entrez le nom d'utilisateur principal de votre cluster Amazon DocumentDB cible.
- Pour Mot de passe, entrez le mot de passe principal de votre cluster Amazon DocumentDB cible.
- Dans Nom de la base de données, entrez le même nom de base de données que celui que vous avez utilisé pour configurer votre point de terminaison source.



4. Testez votre connexion pour vérifier qu'elle a été correctement configurée.



5. Choisissez Créer un point de terminaison.

Étape 7 : créer et exécuter une tâche de migration

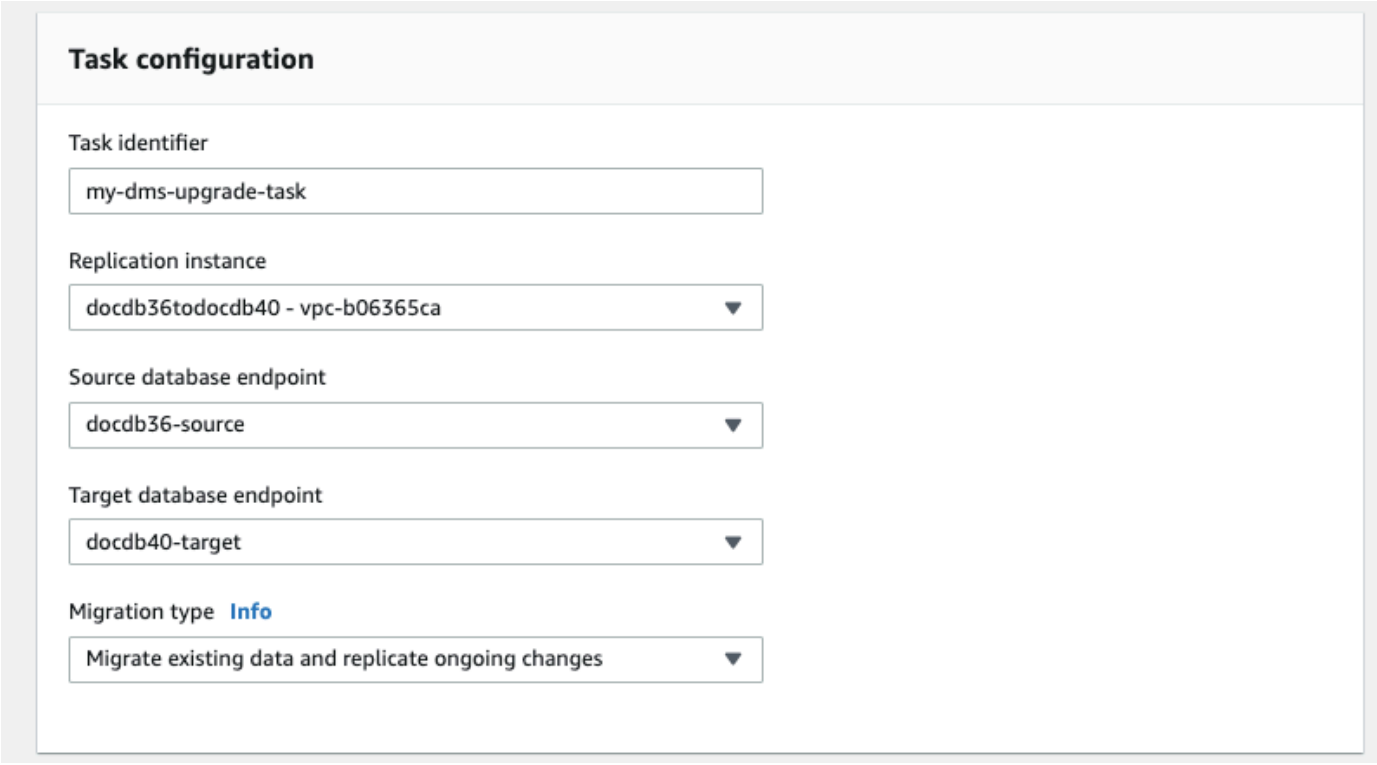
Une AWS DMS tâche lie l'instance de réplication à vos instances source et cible. Lorsque vous créez une tâche de migration, vous spécifiez le point de terminaison source, le point de terminaison cible, l'instance de réplication et tous les paramètres de migration souhaités. Une AWS DMS tâche peut être créée avec trois types de migration différents : migrer des données existantes, migrer des données existantes et répliquer les modifications en cours ou répliquer uniquement les modifications de données. Le but de cette procédure pas à pas étant de mettre à niveau un cluster Amazon DocumentDB avec un temps d'arrêt minimal, les étapes utilisent l'option permettant de migrer les données existantes et de répliquer les modifications en cours. Avec cette option, AWS DMS capture les modifications lors de la migration de vos données existantes. AWS DMS continue de capturer et d'appliquer les modifications même après le chargement des données en masse. Les bases de données source et cible sont finalement synchronisées, permettant ainsi une migration avec un temps d'indisponibilité minimal.

Vous trouverez ci-dessous les étapes à suivre pour créer une tâche de migration afin de minimiser les interruptions de service :

1. Ouvrez la AWS DMS console.
2. Dans le volet de navigation, choisissez Tasks.

3. Choisissez Create task (Créer une tâche) et indiquez les informations suivantes :

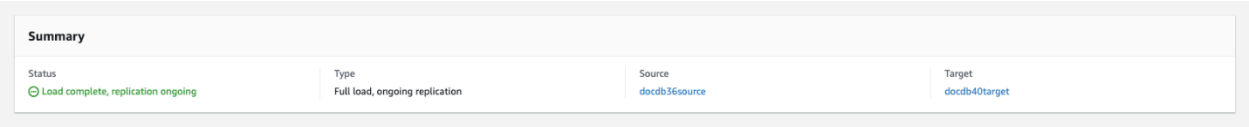
   - Pour Identifiant de tâche, entrez un nom facile à retenir, par exemple`my-dms-upgrade-task`.
   - Pour l'instance de réplication, choisissez l'instance de réplication que vous avez créée à l'étape 3 : Création d'une instance de AWS Database Migration Service réplication
   - Pour Point de terminaison de base de données source, choisissez le point de terminaison source que vous avez créé à l'étape 4 : Création d'un point de terminaison AWS Database Migration Service source
   - Pour le point de terminaison de base de données cible, choisissez le point de terminaison cible que vous avez créé à l'étape 5 : Création d'un point de terminaison AWS Database Migration Service cible
   - Pour le type de migration, choisissez Migrer les données existantes et répliquer les modifications en cours.



4. Dans la section Paramètres des tâches, activez CloudWatch les journaux.
5. Pour la section Mappages de tables, choisissez Ne rien faire. Cela garantira que les index créés à l'étape 3 ne seront pas supprimés.

6. Pour la configuration de démarrage de la tâche de migration, choisissez Automatiquement lors de la création. Cela lancera automatiquement la tâche de migration une fois que vous l'aurez créée.
7. Choisissez Créer tâche.

AWS DMS commence maintenant à migrer les données de votre cluster Amazon DocumentDB source vers votre cluster Amazon DocumentDB cible. L'état de la tâche doit passer de Démarrage à Exécution. Vous pouvez suivre la progression en choisissant Tâches dans la AWS DMS console. Au bout de quelques minutes/heures (en fonction de la taille de votre migration), le statut devrait passer à Chargement terminé, réplication en cours. Cela signifie que vous AWS DMS avez effectué une migration complète de votre cluster Amazon DocumentDB source vers un cluster Amazon DocumentDB cible et que vous répliquez désormais les événements de modification.



Votre source et votre cible finiront par être synchronisées. Vous pouvez vérifier s'ils sont synchronisés en exécutant une `count()` opération sur vos collections pour vérifier que tous les événements de modification ont été migrés.

Étape 8 : remplacement du point de terminaison de l'application par le cluster Amazon DocumentDB cible

Une fois le chargement complet terminé et le processus CDC répliqué en continu, vous êtes prêt à changer le point de terminaison de connexion à la base de données de votre application de votre cluster Amazon DocumentDB source vers votre cluster Amazon DocumentDB cible.

Outils de migration

Pour migrer vers Amazon DocumentDB, les deux principaux outils utilisés par la plupart des clients sont the AWS Database Migration Service (AWS DMS) et les utilitaires de ligne de commande tels que `mongodump` et. `mongorestore` En tant que bonne pratique, et pour l'une ou l'autre de ces options, nous vous recommandons de créer d'abord des index dans Amazon DocumentDB avant de commencer votre migration, car cela peut réduire le temps global et accélérer la migration. Pour ce faire, vous pouvez utiliser l'outil d'indexation Amazon DocumentDB.

AWS Database Migration Service

AWS Database Migration Service (AWS DMS) est un service cloud qui facilite la migration de bases de données relationnelles et non relationnelles vers Amazon DocumentDB. Vous pouvez l'utiliser AWS DMS pour migrer vos données vers Amazon DocumentDB à partir de bases de données hébergées sur site ou sur EC2. Vous pouvez ainsi effectuer des migrations ponctuelles ou répliquer les modifications en cours pour synchroniser les sources et les cibles. AWS DMS

Pour plus d'informations sur l'utilisation AWS DMS de la migration vers Amazon DocumentDB, consultez :

- Utilisation de MongoDB comme source pour AWS DMS
- Utilisation d'Amazon DocumentDB comme cible pour AWS Database Migration Service
- Procédure pas à pas : migration de MongoDB vers Amazon DocumentDB

Utilitaires de ligne de commande

Les utilitaires courants pour la migration de données vers et depuis Amazon DocumentDB `mongodump` incluent`mongorestore`,`mongoexport`, et. `mongoimport` Généralement, `mongodump` et `mongorestore` sont les utilitaires les plus efficaces car ils vident et restaurent les données de vos bases de données dans un format binaire. Il s'agit généralement de l'option la plus performante et génère une taille de données plus petite que les exportations logiques. `mongoexport` et `mongoimport` sont utiles si vous souhaitez exporter et importer des données dans un format logique comme JSON ou CSV, car les données sont lisibles par l'utilisateur ; ils sont toutefois plus lents que `mongodump`/`mongorestore` et génèrent une taille de données plus importante.

La Approches de migration section ci-dessous explique quand il est préférable d'utiliser AWS DMS les utilitaires de ligne de commande en fonction de votre cas d'utilisation et de vos besoins.

Découverte

Pour chacune de vos déploiements MongoDB, vous devez identifier et enregistrer deux jeux de données : Détails d'architecture et Caractéristiques d'exploitation. Ces informations vous permettront de choisir l'approche de migration appropriée et le dimensionnement du cluster.

Détails d'architecture

- Nom

Choisissez un nom unique pour le suivi de ce déploiement.

- Version

  Enregistrez la version de MongoDB que votre déploiement exécute. Pour rechercher la version, connectez-vous à un membre du jeu de réplicas avec le shell Mongo et exécutez l'opération `db.version()`.

- Type

  Enregistrez si votre déploiement est une instance mongo autonome, un jeu de réplicas ou un cluster partitionné.

- Membres

  Enregistrez les noms d'hôte, les adresses, et les ports de chaque cluster, jeu de réplicas ou membre autonome.

  Pour un déploiement en cluster, vous pouvez trouver les membres de partition mongo en vous connectant à un hôte avec le shell Mongo et en exécutant l'opération `sh.status()`.

  Pour un jeu de réplicas, vous pouvez obtenir les membres en vous connectant à un membre de l'ensemble de réplicas avec le shell Mongo et en exécutant l'opération `rs.status()`.

- Tailles oplog

  Pour les jeux de réplicas ou les clusters partitionnés, enregistrez la taille de l'oplog pour chaque membre de l'ensemble de réplicas. Pour rechercher la taille oplog d'un membre, connectez-vous à un membre de l'ensemble de réplicas avec le shell mongo et exécutez l'opération `ps.printReplicationInfo()`.

- Priorités des membres de l'ensemble de réplicas

  Pour les jeux de réplicas ou les clusters partitionnés, enregistrez la priorité de chaque membre du jeu de réplicas. Pour rechercher les priorités des membres du jeu de réplicas, connectez-vous à un membre du jeu de réplicas avec le shell Mongo et exécutez l'opération `rs.conf()`. La priorité est affichée en tant que valeur de la clé `priority`.

- Utilisation de TLS/SSL

  Enregistrez si le protocole TLS (Transport Layer Security)/Secure Sockets Layer (SSL) est utilisé sur chaque nœud pour le chiffrement en transit.

Caractéristiques d'exploitation

- Statistiques de base de données

  Pour chaque collection, enregistrez les informations suivantes :

  - Nom
  - La taille des données
  - Nombre de collections

  Pour rechercher les statistiques de la base de données, connectez-vous à cette dernière avec le shell Mongo et exécutez la commande `db.runCommand({dbstats: 1})`.

- Statistiques de collection

  Pour chaque collection, enregistrez les informations suivantes :

  - Espace de noms
  - La taille des données
  - Nombre d'index
  - Si la collection est limitée

- Statistiques d'index

  Pour chaque collection, enregistrez les informations d'index suivantes :

  - Espace de noms
  - ID
  - Size
  - Clés
  - TTL
  - Fragmentée
  - Contexte

  Pour rechercher les informations d'index, connectez-vous à la base de données avec le shell Mongo et exécutez la commande `db.collection.getIndexes()`.

- Compteurs d'opérations

  Ces informations vous permettent de comprendre vos modèles de charge de travail MongoDB actuels (beaucoup de lectures, beaucoup d'écritures ou équilibre). Il fournit également des conseils sur votre sélection initiale d'instance Amazon DocumentDB.

  Voici les informations à collecter au cours de la période de surveillance (en nombre/sec) :

  - Requêtes
  - Insertions
  - Mises à jour
  - Suppressions

  Vous pouvez obtenir ces informations en représentant dans un graphique le résultat de la commande `db.serverStatus()` au fil du temps. Vous pouvez également utiliser l'outil

mongostat pour obtenir des valeurs instantanées pour ces statistiques. Toutefois, avec cette option, vous risquez de planifier votre migration sur des périodes d'utilisation autres que vos pics de charge.

- Statistiques réseau

  Ces informations vous permettent de comprendre vos modèles de charge de travail MongoDB actuels (beaucoup de lectures, beaucoup d'écritures ou équilibre). Il fournit également des conseils sur votre sélection initiale d'instance Amazon DocumentDB.

  Voici les informations à collecter au cours de la période de surveillance (en nombre/sec) :

  - Connexions
  - Octets réseau entrants
  - Octets réseau sortants

  Vous pouvez obtenir ces informations en représentant dans un graphique le résultat de la commande `db.serverStatus()` au fil du temps. Vous pouvez également utiliser l'outil mongostat pour obtenir des valeurs instantanées pour ces statistiques. Toutefois, avec cette option, vous risquez de planifier votre migration sur des périodes d'utilisation autres que vos pics de charge.

Planification : exigences du cluster Amazon DocumentDB

Une migration réussie nécessite que vous examiniez attentivement à la fois la configuration de votre cluster Amazon DocumentDB et la manière dont les applications accèderont à votre cluster. Réfléchissez à chacune des dimensions suivantes pour déterminer les exigences de votre cluster :

- Disponibilité

  Amazon DocumentDB fournit une haute disponibilité grâce au déploiement d'instances de réplication, qui peuvent être promues au rang d'instance principale dans le cadre d'un processus appelé failover. En déployant des instances de réplica dans différentes zones de disponibilité, vous pouvez atteindre des niveaux de disponibilité plus élevés.

Le tableau suivant fournit des directives relatives aux configurations de déploiement d'Amazon DocumentDB afin de répondre à des objectifs de disponibilité spécifiques.

Objectif de disponibilité	Total des instances	Réplicas	Zones de disponibilité
99 %	1	0	1
99,9 %	2	1	2
99,99 %	3	2	3

La fiabilité globale du système doit prendre en compte tous les composants, pas seulement la base de données. Pour connaître les meilleures pratiques et les recommandations visant à répondre aux besoins globaux de fiabilité du système, consultez le livre blanc AWS Well-Architected Reliability Pillar.

- Performances

  Les instances Amazon DocumentDB vous permettent de lire et d'écrire sur le volume de stockage de votre cluster. Les instances de cluster peuvent avoir différents types, avec différentes tailles de mémoire et de vCPU, ce qui affecte les performances en lecture et en écriture de votre cluster. À l'aide des informations que vous avez collectées lors de la phase de détection, choisissez un type d'instance capable de prendre en charge vos exigences de performances pour la charge de travail. Pour obtenir la liste des types d’instances, consultez Gestion de classes d'instance.

  Lorsque vous choisissez un type d'instance pour votre cluster Amazon DocumentDB, tenez compte des aspects suivants des exigences de performance de votre charge de travail :

- vCPU : les architectures qui nécessitent un nombre de connexions plus élevé peuvent tirer parti des instances dotées d'un plus grand nombre de vCPU.

- Mémoire : lorsque cela est possible, le fait de conserver votre ensemble de données de travail en mémoire garantit des performances optimales. Une règle de départ consiste à réserver un tiers de la mémoire de votre instance pour le moteur Amazon DocumentDB, en laissant les deux tiers pour votre ensemble de données de travail.

- Connexions — Le nombre de connexions optimal minimum est de huit connexions par vCPU d'instance Amazon DocumentDB. Bien que la limite de connexion aux instances Amazon DocumentDB soit beaucoup plus élevée, les avantages en termes de performances liés aux connexions supplémentaires diminuent au-delà de huit connexions par vCPU.

- Réseau : les charges de travail comportant un grand nombre de clients ou de connexions doivent tenir compte des performances réseau agrégées requises pour les données insérées et extraites. Les opérations en bloc peuvent utiliser plus efficacement les ressources réseau.

- Performances d'insertion : les insertions de documents uniques constituent généralement le moyen le plus lent d'insérer des données dans Amazon DocumentDB. Les opérations d'insertion en bloc peuvent être nettement plus rapides que les insertions uniques.

- Performances de lecture : les lectures depuis la mémoire de travail sont toujours plus rapides que les lectures renvoyées depuis le volume de stockage. Par conséquent, l'optimisation de la taille de la mémoire d'instance afin de conserver votre jeu de travail en mémoire est idéale.

En plus de servir les lectures depuis votre instance principale, les clusters Amazon DocumentDB sont automatiquement configurés en tant que jeux de répliques. Vous pouvez alors acheminer les requêtes en lecture seule vers les réplicas en lecture en définissant la préférence de lecture dans votre pilote MongoDB. Vous pouvez dimensionner le trafic en lecture en ajoutant des réplicas, ce qui réduit la charge globale sur l'instance principale.

Il est possible de déployer des répliques Amazon DocumentDB de différents types d'instances dans le même cluster. Un exemple de cas d'utilisation peut être le maintien d'un réplica avec un type d'instance plus important pour servir le trafic d'analyses temporaire. Si vous déployez un ensemble de types d'instance varié, assurez-vous de configurer la priorité de basculement pour chaque instance. Vous vous assurerez ainsi qu'un événement de basculement promeut toujours un réplica d'une taille suffisante pour gérer votre charge d'écriture.

- Récupération

  Amazon DocumentDB sauvegarde en permanence vos données au fur et à mesure qu'elles sont écrites. Il fournit des fonctionnalités de point-in-time restauration (PITR) sur une période configurable de 1 à 35 jours, connue sous le nom de période de rétention des sauvegardes. La durée de conservation des sauvegardes par défaut est d'un jour. Amazon DocumentDB crée également automatiquement des instantanés quotidiens de votre volume de stockage, qui sont également conservés pendant la période de conservation des sauvegardes configurée.

  Si vous souhaitez conserver les instantanés au-delà de la période de conservation des sauvegardes, vous pouvez également lancer des instantanés manuels à tout moment à l'aide du bouton AWS Management Console et AWS Command Line Interface ()AWS CLI. Pour plus d’informations, consultez Sauvegarde et restauration dans Amazon DocumentDB.

  Tenez compte des éléments suivants lorsque vous planifiez votre migration :

  - Choisissez une période de conservation des sauvegardes de 1 à 35 jours qui répond à votre objectif de point de restauration (RPO).
  - Déterminez si vous avez besoin que les instantanés soient créés manuellement et, le cas échéant, à quel intervalle.

Approches de migration

Il existe trois approches principales pour migrer vos données vers Amazon DocumentDB.

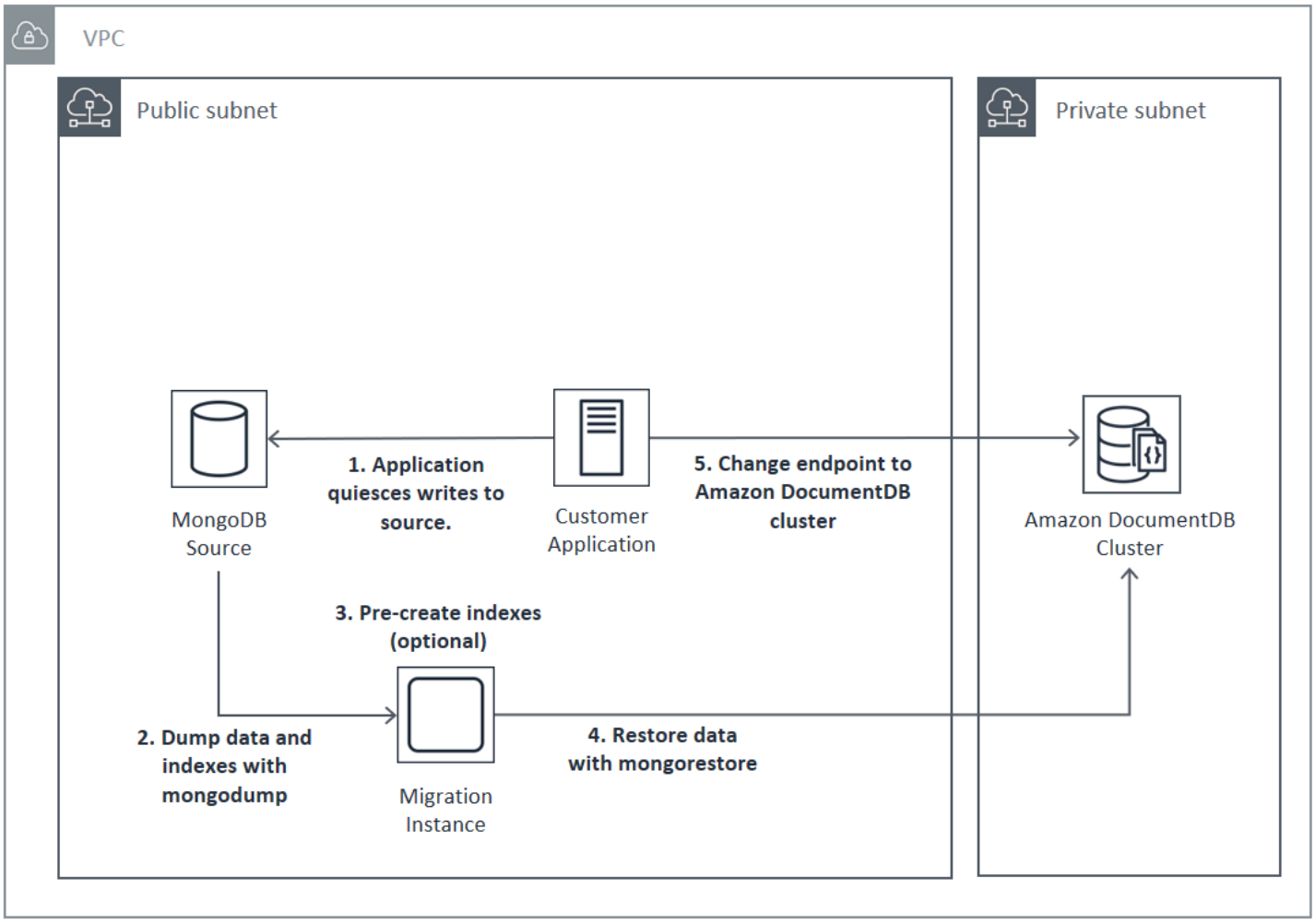
> **Note**
>
> Bien que vous puissiez créer des index à tout moment dans Amazon DocumentDB, il est globalement plus rapide de créer vos index avant d'importer des ensembles de données volumineux. À titre de bonne pratique, nous vous recommandons, pour chacune des approches ci-dessous, de créer d'abord vos index dans Amazon DocumentDB avant d'effectuer la migration. Pour ce faire, vous pouvez utiliser l'outil d'indexation Amazon DocumentDB.

Hors connexion

L'approche hors ligne utilise les `mongorestore` outils `mongodump` et pour migrer vos données de votre déploiement MongoDB source vers votre cluster Amazon DocumentDB. La méthode hors ligne est l'approche de migration la plus simple, mais elle entraîne aussi le plus de temps d'arrêt pour votre cluster.

Le processus de base pour la migration hors ligne se présente comme suit :

1. Arrêtez les écritures sur votre source MongoDB.
2. Videz les index et les ensembles de données du déploiement MongoDB source.
3. Si vous migrez vers un cluster élastique, créez vos collections partitionnées à l'aide de la `sh.shardCollection()` commande. Si vous migrez vers un cluster basé sur une instance, passez à l'étape suivante.
4. Restaurez les index dans le cluster Amazon DocumentDB.
5. Restaurez les données de collecte sur le cluster Amazon DocumentDB.
6. Modifiez le point de terminaison de votre application pour écrire dans le cluster Amazon DocumentDB.



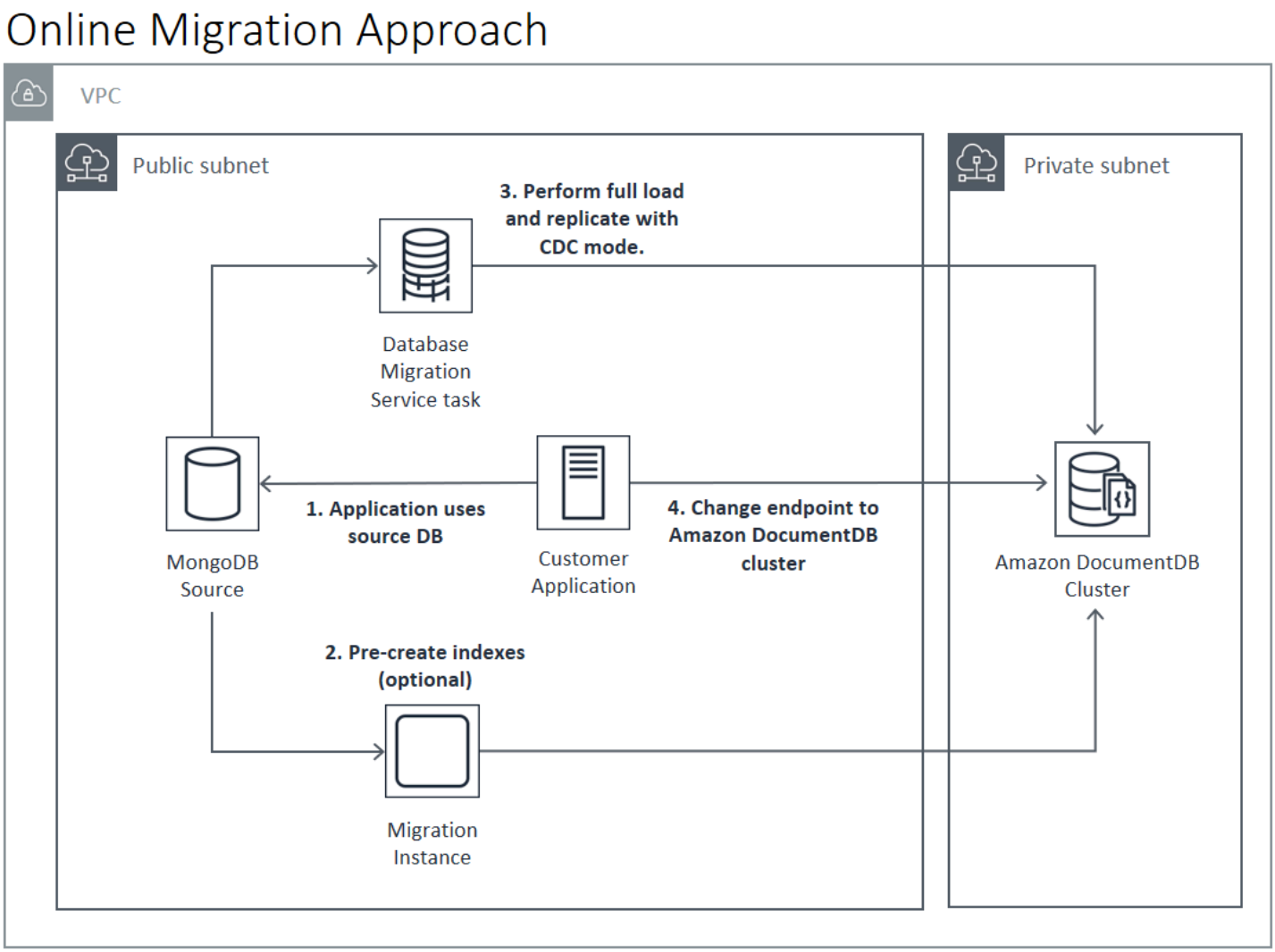
En ligne

L'approche en ligne utilise AWS Database Migration Service (AWS DMS). Il effectue un chargement complet des données de votre déploiement MongoDB source vers votre cluster Amazon DocumentDB. Il bascule ensuite en mode de capture des données modifiées (CDC) pour répliquer les modifications. L'approche en ligne limite les temps d'arrêt pour votre cluster, mais cette méthode est la plus lente des trois.

Le processus de base pour la migration en ligne se présente comme suit :

1. Votre application utilise la base de données source normalement.
2. Si vous migrez vers un cluster élastique, créez vos collections partitionnées à l'aide de la `sh.shardCollection()` commande. Si vous migrez vers un cluster basé sur une instance, passez à l'étape suivante.

3. Créez au préalable des index dans le cluster Amazon DocumentDB.
4. Créez une AWS DMS tâche pour effectuer un chargement complet, puis activez le CDC depuis le déploiement source de MongoDB vers le cluster Amazon DocumentDB.
5. Une fois le chargement complet de la AWS DMS tâche terminé et la réplication des modifications apportées à Amazon DocumentDB, basculez le point de terminaison de l'application vers le cluster Amazon DocumentDB.



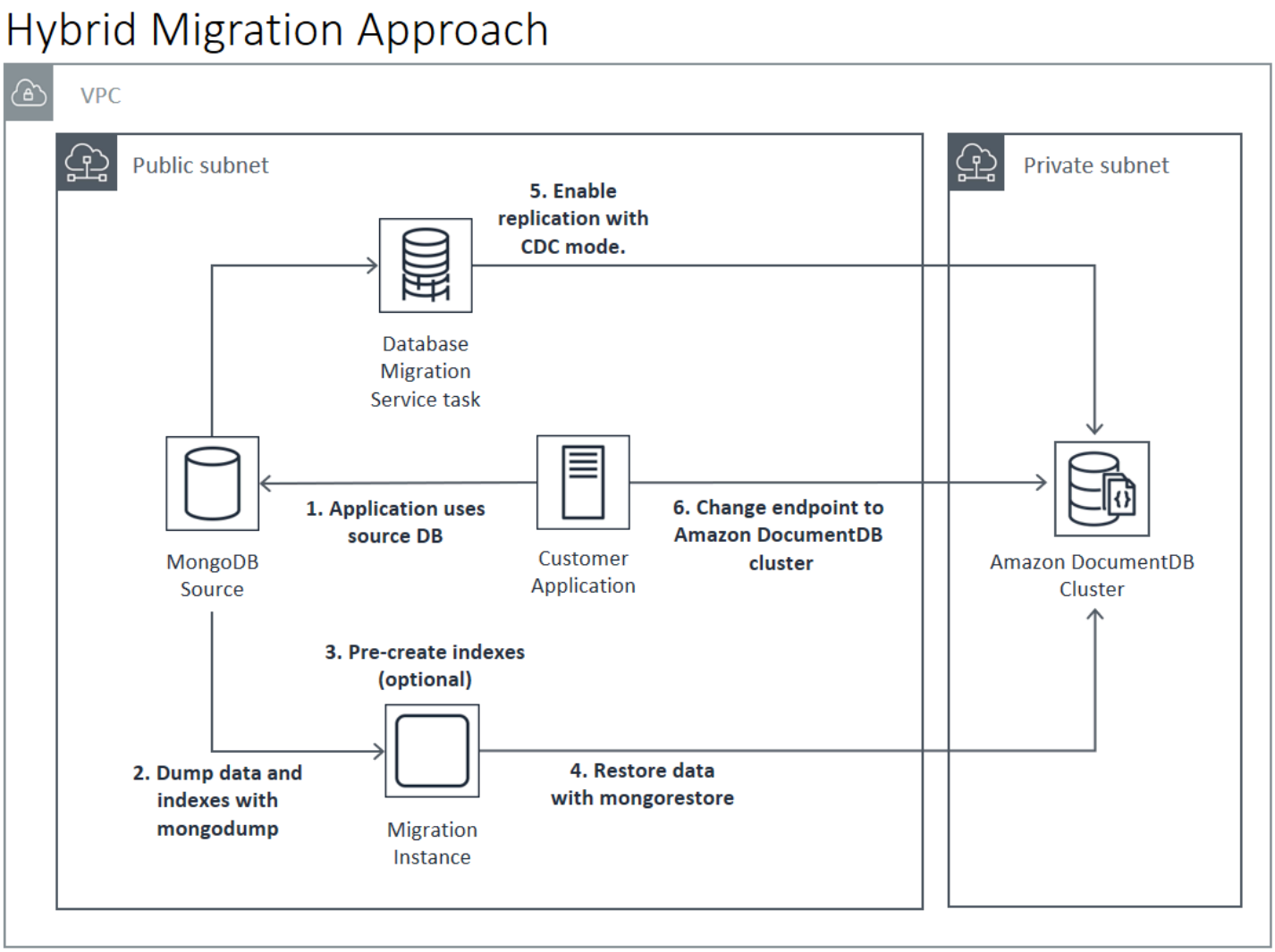
Pour plus d'informations sur l'utilisation AWS DMS pour migrer, consultez la section Utilisation d'Amazon DocumentDB comme cible pour AWS Database Migration Service et le didacticiel associé dans le guide de l'AWS Database Migration Service utilisateur.

Hybride

L'approche hybride utilise les `mongorestore` outils `mongodump` et pour migrer vos données de votre déploiement MongoDB source vers votre cluster Amazon DocumentDB. Il est ensuite utilisé AWS DMS en mode CDC pour répliquer les modifications. L'approche hybride équilibre la vitesse de migration et les temps d'arrêt, mais constitue l'approche la plus complexe des trois.

Le processus de base pour la migration hybride se présente comme suit :

1. Votre application utilise le déploiement MongoDB source normalement.
2. Videz les index et les ensembles de données du déploiement MongoDB source.
3. Restaurez les index dans le cluster Amazon DocumentDB.
4. Si vous migrez vers un cluster élastique, créez vos collections partitionnées à l'aide de la `sh.shardCollection()` commande. Si vous migrez vers un cluster basé sur une instance, passez à l'étape suivante.
5. Restaurez les données de collecte sur le cluster Amazon DocumentDB.
6. Créez une AWS DMS tâche pour activer le CDC depuis le déploiement source de MongoDB vers le cluster Amazon DocumentDB.
7. Lorsque la AWS DMS tâche réplique les modifications dans une fenêtre acceptable, modifiez le point de terminaison de votre application pour écrire dans le cluster Amazon DocumentDB.



> **Important**
>
> Une AWS DMS tâche ne peut actuellement migrer qu'une seule base de données. Si votre source MongoDB comporte un grand nombre de bases de données, vous devrez peut-être automatiser la création des tâches de migration ou envisager d'utiliser la méthode hors connexion.

Quelle que soit l'approche de migration que vous choisissez, il est plus efficace de pré-créer des index dans votre cluster Amazon DocumentDB avant de migrer vos données. Cela est dû au fait que les index Amazon DocumentDB sont des données insérées en parallèle, mais la création d'un index sur des données existantes est une opération monothread.

Comme il AWS DMS ne migre pas les index (uniquement vos données), aucune étape supplémentaire n'est requise pour éviter de créer des index une deuxième fois.

Sources de migration

Si votre source MongoDB est un processus mongo autonome et que vous souhaitez utiliser les approches de migration en ligne ou hybride, vous devez d'abord convertir votre mongo autonome en jeu de réplicas afin que l'oplog soit créé en vue d'une utilisation en tant que source CDC.

Si vous migrez à partir d'un cluster partitionné ou de jeux de réplicas MongoDB, envisagez de créer un secondaire enchainé ou masqué pour chaque jeu de réplicas ou partition à utiliser comme source de migration. L'exécution des vidages des données peut forcer l'utilisation des jeux de données hors de la mémoire et avoir un impact sur les performances des instances de production. Vous pouvez réduire ce risque en migrant à partir d'un nœud qui ne sert pas des données de production.

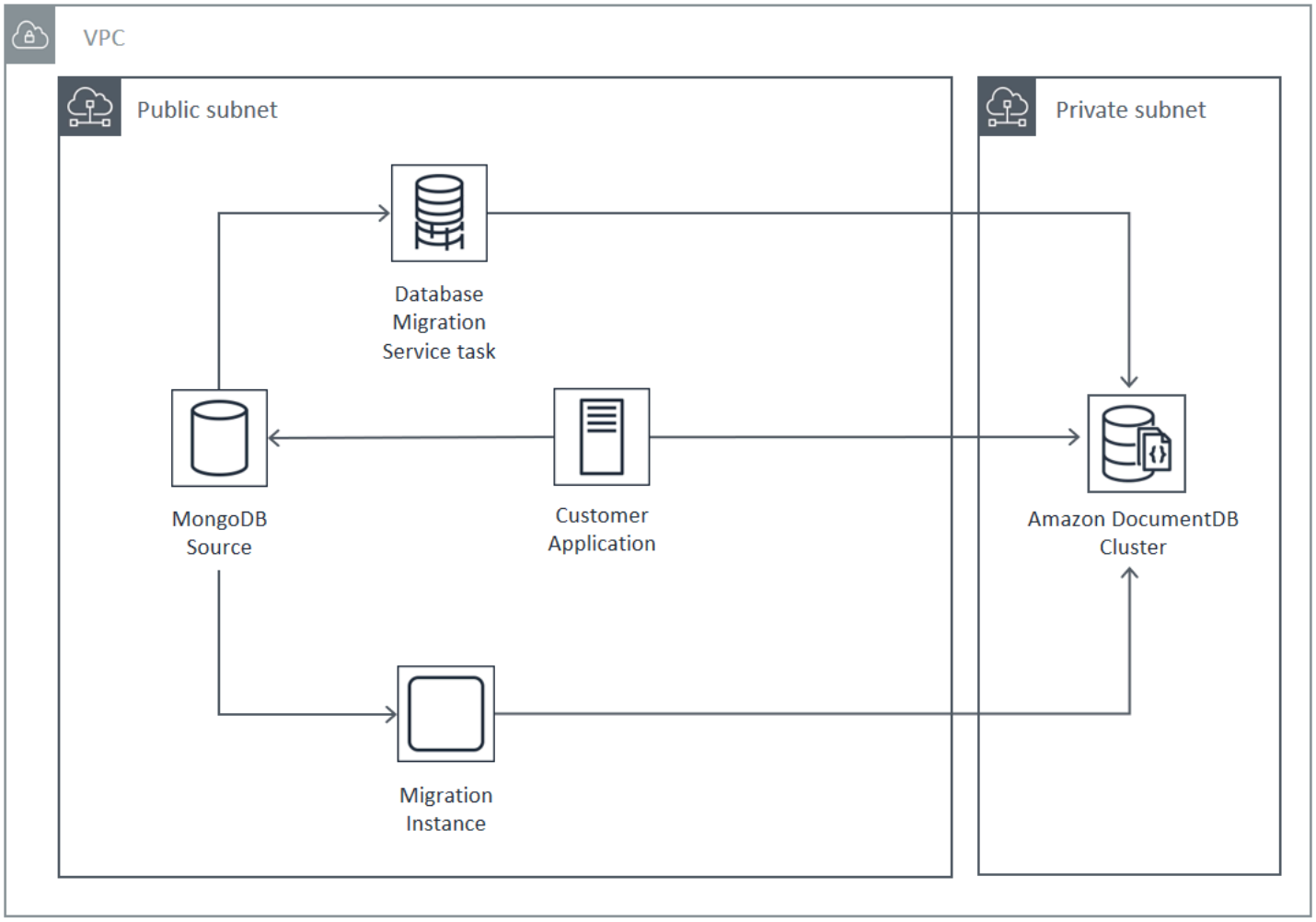
Versions des sources de migration

Si la version de votre base de données MongoDB source est différente de la version de compatibilité de votre cluster Amazon DocumentDB de destination, vous devrez peut-être prendre d'autres mesures de préparation pour garantir une migration réussie. Les deux exigences les plus courantes sont la nécessité de mettre à niveau l'installation source de MongoDB vers une version prise en charge pour la migration (MongoDB version 3.0 ou supérieure) et la mise à niveau des pilotes de votre application pour prendre en charge la version cible d'Amazon DocumentDB.

Si votre migration possède l'une ou l'autre de ces exigences, veillez à intégrer ces étapes de votre plan de migration pour mettre à niveau et tester toutes les modifications du pilote.

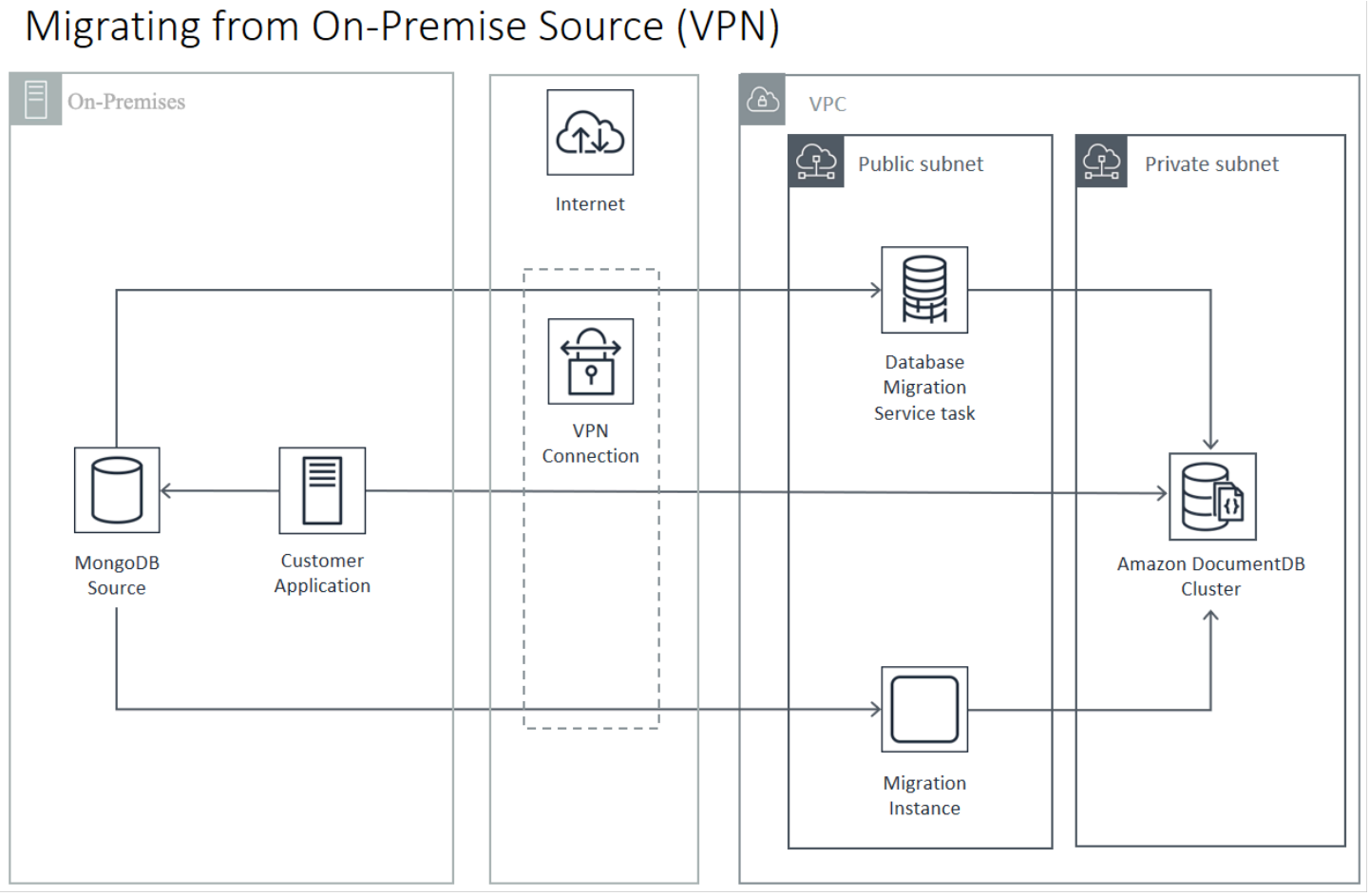
Connectivité de la migration

Vous pouvez migrer vers Amazon DocumentDB depuis un déploiement MongoDB source exécuté dans votre centre de données ou depuis un déploiement MongoDB exécuté sur une instance Amazon EC2. La migration à partir de MongoDB exécuté sur EC2 est simple et exige seulement une configuration correcte de vos groupes et sous-réseaux de sécurité.

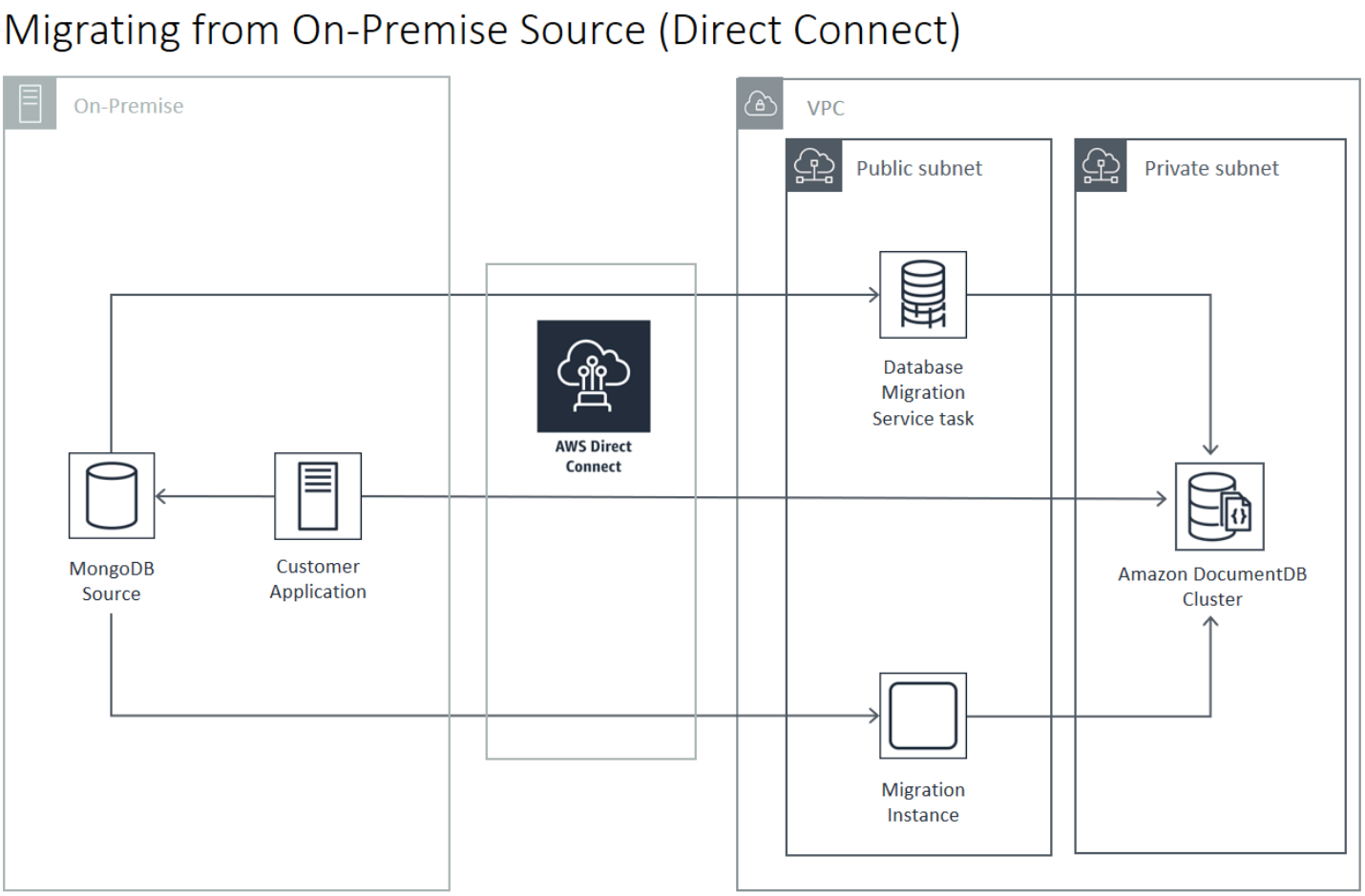


La migration à partir d'une base de données locale requiert une connexion entre votre déploiement MongoDB et votre cloud privé virtuel (VPC). Vous pouvez y parvenir par le biais d'une connexion à un réseau privé virtuel (VPN) ou en utilisant le AWS Direct Connect service. Même si vous pouvez migrer via Internet vers votre VPC, cette méthode de connexion est la moins souhaitable du point de vue de la sécurité.

Le schéma suivant illustre une migration vers Amazon DocumentDB depuis une source sur site via une connexion VPN.



Ce qui suit représente une migration vers Amazon DocumentDB à partir d'une source locale à l'aide de. AWS Direct Connect



Les approches de migration en ligne et hybrides nécessitent l'utilisation d'une AWS DMS instance, qui doit s'exécuter sur Amazon EC2 dans un Amazon VPC. Toutes les approches nécessitent que le serveur de migration exécute `mongodump` et `mongorestore`. Il est généralement plus facile d'exécuter le serveur de migration sur une instance Amazon EC2 dans le VPC où votre cluster Amazon DocumentDB est lancé, car cela simplifie considérablement la connectivité à votre cluster Amazon DocumentDB.

Test

Voici des objectifs de test prémigration :

- Vérifiez que l'approche choisie donne le résultat de migration souhaité.
- Vérifiez que vos choix de type d'instance et préférences de lecture respectent les exigences en matière de performances de votre application.
- Vérifiez le comportement de votre application lors du basculement.

Réflexions sur le test du plan de migration

Tenez compte des points suivants lorsque vous testez votre plan de migration Amazon DocumentDB.

Rubriques

Restauration des index

Par défaut, `mongorestore` crée des index pour les collections vidées, mais il les crée une fois que les données ont été restaurées. Dans l'ensemble, il est plus rapide de créer des index dans Amazon DocumentDB avant que les données ne soient restaurées dans le cluster. En effet, les opérations d'indexation sont mises en parallèle pendant le chargement des données.

Si vous choisissez de précréer vos index, vous pouvez ignorer l'étape de création d'index lors de la restauration des données avec `mongorestore` en fournissant l'option `--noIndexRestore`.

Vidage de données

L'outil `mongodump` est la méthode préférée de vidage des données à partir de votre déploiement MongoDB source. Selon les ressources disponibles sur votre instance de migration, vous pouvez être en mesure d'accélérer votre `mongodump` en augmentant le nombre de connexions parallèles vidées à partir de la valeur par défaut 4 à l'aide de l'option `--numParallelCollections`.

Restauration des données

`mongorestore`Cet outil est la méthode préférée pour restaurer les données transférées sur votre instance Amazon DocumentDB. Vous pouvez améliorer les performances de restauration en augmentant le nombre d'agents de travail pour chaque collection pendant la restauration avec l'option `--numInsertionWorkersPerCollection`. Un travailleur par vCPU sur votre instance principale de cluster Amazon DocumentDB est un bon point de départ.

Amazon DocumentDB ne prend actuellement pas en charge l'option de l'`mongorestore--oplogReplay`outil.

Par défaut, `mongorestore` ignore les erreurs d'insertion et continue le processus de restauration. Cela peut se produire si vous restaurez des données non prises en charge sur votre instance Amazon DocumentDB. Par exemple, cela peut se produire si vous avez un document qui contient des valeurs ou clés avec des chaînes null. Si vous préférez que l'opération `mongorestore` échoue totalement si une erreur de restauration se produit, utilisez l'option `--stopOnError`.

Dimensionnement Oplog

Le journal des opérations MongoDB (`oplog`) est une collection limitée qui contient toutes les modifications apportées à votre base de données. Vous pouvez afficher la taille de l'oplog et la plage de temps qu'il contient en exécutant l'opération db`.printReplicationInfo()` sur un jeu de réplicas ou un membre de partition.

Si vous utilisez des approches en ligne ou hybrides, assurez-vous que le journal journal de chaque jeu de répliques ou de chaque partition est suffisamment grand pour contenir toutes les modifications apportées pendant toute la durée du processus de migration des données (que ce soit par le biais du chargement complet d'une tâche `mongodump` ou d'une AWS DMS tâche), ainsi qu'une mémoire tampon raisonnable. Pour de plus amples informations, veuillez consulter Vérification de la taille du Oplog dans la documentation MongoDB. Déterminez la taille oplog minimale requise en enregistrant le temps écoulé pour la première série de tests de votre processus `mongodump` ou `mongorestore`, ou de la tâche de chargement complet AWS DMS .

AWS Database Migration Service Configuration

Le guide de AWS Database Migration Service l'utilisateur décrit les composants et les étapes nécessaires à la migration de vos données sources MongoDB vers votre cluster Amazon DocumentDB. Le processus de base AWS DMS à utiliser pour effectuer une migration en ligne ou hybride est le suivant :

Pour effectuer une migration à l'aide de AWS DMS

1. Créez un point de terminaison source MongoDB. Pour plus d'informations, consultez Utilisation de MongoDB comme source pour AWS DMS.
2. Créez un point de terminaison cible Amazon DocumentDB. Pour plus d'informations, consultez Utilisation des points de terminaison AWS DMS.

Si vous configurez votre point de terminaison cible en tant que cluster élastique, notez que votre certificat SSL Amazon DocumentDB existant ne fonctionnera pas avec les clusters élastiques et que vous devrez attacher un nouveau certificat SSL à votre point de terminaison en suivant les étapes suivantes :

a. Rendez-vous sur https://www.amazontrust.com/repository/SFSRootCAG2.pem et enregistrez le contenu dans un fichier « SFSRootCag2.pem ». Il s'agit du fichier de certificat que vous devrez importer lors des étapes suivantes.

b. Lors de la création du point de terminaison du cluster élastique, sous Configuration du point de terminaison, choisissez Ajouter un nouveau certificat CA.

- Pour Identifiant de certificat, entrez `SFSRootCAG2.pem`.
- Dans Import certificate file (Importer un fichier de certificat), choisissez Choose file (Choisir un fichier) et accédez au fichier `SFSRootCAG2.pem` que vous avez téléchargé précédemment. Sélectionnez et ouvrez le fichier. Choisissez Importer un certificat, puis `SFSRootCAG2.pem` choisissez dans le menu déroulant Choisir un certificat.

3. Créez au moins une instance AWS DMS de réplication. Pour plus d'informations, consultez la section Utilisation d'une instance de AWS DMS réplication.
4. Créez au moins une tâche AWS DMS de réplication. Pour plus d'informations, consultez Utilisation des tâches AWS DMS.

   Pour une migration en ligne, votre tâche de migration utilise le type de migration Migration des données existantes et réplication des modifications continues.

   Pour une migration hybride, votre tâche de migration utilise le type de migration Replicate data changes only (Répliquer les changements de données uniquement). Vous pouvez choisir l'heure de début CDC pour vous aligner sur le temps de vidage de votre opération `mongodump`. L'oplog MongoDB est idempotent. Pour éviter de passer à côté de modifications, il est conseillé de conserver quelques minutes de chevauchement entre votre heures de fin `mongodump` et votre heure de début CDC.

Migration à partir d'un cluster partitionné

Le processus de migration des données d'un cluster fragmenté MongoDB vers votre instance Amazon DocumentDB est essentiellement celui de plusieurs migrations de répliques en parallèle. Dans le cadre du test de la migration d'un cluster partitionné, il est essentiel de tenir compte du fait

que certaines partitions peuvent être plus largement utilisées que d'autres. Cette situation entraîne différents délais pour la migration de données. Assurez-vous d'évaluer les `oplog` exigences de chaque partition lors de la planification et des tests.

Voici quelques problèmes de configuration à prendre en compte lors de la migration d'un cluster partitionné :

- Avant d'exécuter `mongodump` ou de démarrer une tâche de migration AWS DMS , vous devez désactiver l'équilibreur de cluster partitionné et attendre la fin de toutes les migrations en cours. Pour de plus amples informations, veuillez consulter Désactiver l'équilibreur dans la documentation MongoDB.
- Si vous utilisez AWS DMS pour répliquer des données, exécutez la `cleanupOrphaned` commande sur chaque partition avant d'exécuter les tâches de migration. Si vous n'exécutez pas cette commande, les tâches risquent d'échouer en raison d'ID de document en double. Notez que cette commande peut avoir un impact sur les performances. Pour de plus amples informations, veuillez consulter cleanupOrphaned dans la documentation MongoDB.
- Si vous utilisez l'outil `mongodump` pour vider les données, vous devez exécuter un processus `mongodump` par partition. L'approche la plus efficace en terme de durée peut nécessiter plusieurs serveurs de migration pour optimiser vos performances de vidage.
- Si vous utilisez AWS Database Migration Service pour répliquer des données, vous devez créer un point de terminaison source pour chaque partition. Exécutez également au moins une tâche de migration pour chaque partition que vous migrez. L'approche la plus efficace en terme de durée peut nécessiter plusieurs instances de réplication pour optimiser vos performances de migration.

Tests de performance

Après avoir migré avec succès vos données vers votre cluster Amazon DocumentDB de test, exécutez votre charge de travail de test sur le cluster. Vérifiez à l'aide CloudWatch des métriques Amazon que vos performances atteignent ou dépassent le débit actuel de votre déploiement de source MongoDB.

Vérifiez les indicateurs clés d'Amazon DocumentDB suivants :

- Débit réseau
- Write throughput
- Read throughput

- Replica lag

Pour plus d'informations, consultez Surveillance Amazon DocumentDB.

Test du basculement

Vérifiez que le comportement de votre application lors d'un événement de basculement d'Amazon DocumentDB répond à vos exigences de disponibilité. Pour lancer un basculement manuel d'un cluster Amazon DocumentDB sur la console, sur la page Clusters, choisissez l'action Failover dans le menu Actions.

Vous pouvez également lancer un basculement en exécutant l'opération `failover-db-cluster` depuis l' AWS CLI. Pour plus d'informations, consultez `failover-db-cluster`la section Amazon DocumentDB de la AWS CLI référence.

Ressources supplémentaires

Consultez les rubriques suivantes dans le Guide de l'utilisateur AWS Database Migration Service :

- Utilisation d'Amazon DocumentDB comme cible pour AWS Database Migration Service
- Procédure : Migration de MongoDB vers Amazon DocumentDB

Manuel de migration : MongoDB vers Amazon DocumentDB

Ce manuel de migration fournit des ressources et des étapes pour vous aider à migrer d'une base de données MongoDB vers Amazon DocumentDB.

Processus de migration

Vous trouverez ci-dessous les étapes de haut niveau généralement nécessaires à la migration de vos données d'une base de données MongoDB vers Amazon DocumentDB.

Rubriques

- Étape 1 : Compatibilité et différences fonctionnelles
- Étape 2 : Preuve de concept
- Étape 3 : migrer les données

- Étape 4 : Validation des données
- Étape 5 : transfert de candidature

Étape 1 : Compatibilité et différences fonctionnelles

Amazon DocumentDB interagit avec les API MongoDB 3.6, 4.0 et 5.0 open source d'Apache 2.0. Par conséquent, vous pouvez utiliser les mêmes pilotes, applications et outils MongoDB avec Amazon DocumentDB avec peu ou pas de modifications.

La première étape consiste à vérifier la compatibilité entre les opérateurs et les index utilisés par votre application dans votre base de données MongoDB et leur disponibilité dans Amazon DocumentDB, ainsi qu'à comprendre les différences fonctionnelles entre eux.

Compatibilité avec les opérateurs

Utilisez l'outil de compatibilité Amazon DocumentDB* pour découvrir facilement si votre application utilise des opérateurs non pris en charge dans ses requêtes. Cet outil peut analyser les fichiers journaux de votre serveur de base de données MongoDB ou le code source de votre application pour fournir un rapport sur les opérateurs non pris en charge. Si vous constatez l'utilisation d'opérateurs non pris en charge, vous devez modifier votre application pour contourner les opérateurs non pris en charge.

Pour vérifier la compatibilité entre les opérateurs MongoDB utilisés dans votre configuration et les opérateurs Amazon DocumentDB pris en charge, exécutez ce qui suit :

```
git clone https://github.com/awslabs/amazon-documentdb-tools.git
cd amazon-documentdb-tools/compat-tool/
python3 compat.py --version <Amazon DocumentDB version> --directory <mongodb logfiles/
source code>
```

Pour plus d’informations, consultez API MongoDB, opérations et types de données pris en charge.

* Non officiellement pris en charge parAWS.

Compatibilité des index

Vous pouvez utiliser l'outil d'indexation Amazon DocumentDB* pour savoir si vous utilisez des types d'index non pris en charge dans Amazon DocumentDB. Cet outil a besoin d'une connexion à votre base de données source pour lire les définitions d'index.

Pour cela, vous devez d'abord déposer les définitions d'index dans un répertoire à l'aide de l'`--dump-indexes`option. Exécutez ensuite l'outil avec l'`--show-issues`option, en fournissant le répertoire pour localiser les index incompatibles.

Indexes d'exportation :

```
git clone https://github.com/awslabs/amazon-documentdb-tools.git
sudo pip install -r amazon-documentdb-tools/index-tool/requirements.txt
mkdir <directory to dump index definitions>
python3 migrationtools/documentdb_index_tool.py --dump-indexes --dir <directory> --uri
 <source-mongodb-uri>
```

Vérifiez la présence d'index incompatibles :

```
python3 migrationtools/documentdb_index_tool.py --show-issues --dir <dumped-index-
definitions-directory>
```

Si vous constatez l'utilisation de types d'index non pris en charge, vous devez modifier votre application ou votre modèle de données pour contourner ou continuer sans les index incompatibles.

Pour plus d'informations sur les types et propriétés d'index pris en charge dans Amazon DocumentDB, consultez Propriétés de l'index et des index la section Comment indexer sur Amazon DocumentDB.

* Non officiellement pris en charge parAWS.

Différences fonctionnelles

Passez Différences fonctionnelles avec MongoDB en revue pour vous familiariser avec les différences.

Étape 2 : Preuve de concept

Réalisez une preuve de concept en exécutant votre application ou votre suite de tests habituelle sur Amazon DocumentDB pour tester les fonctionnalités et les performances. Vous devrez peut-être renseigner votre cluster Amazon DocumentDB avec des données pour effectuer les tests. Par exemple, vous pouvez utiliser les `mongorestore` outils `mongodump` et pour copier des données depuis votre MongoDB source.

Tests fonctionnels

Créez un cluster Amazon DocumentDB (voirCréation d'un cluster Amazon DocumentDB) et exécutez votre application ou votre suite de tests fonctionnels pour vérifier si tous les flux de travail des applications continuent de fonctionner correctement sur Amazon DocumentDB.

Tests de performance

Exécutez des tests de performances sur votre application ou suite de tests de performances exécutée sur Amazon DocumentDB avec une charge de travail similaire à votre charge de travail de production pour vérifier si la configuration répond à vos exigences de latence. Ajustez votre charge de travail en termes de performances ou adaptez votre cluster Amazon DocumentDB, le cas échéant. Pour plus d'informations, consultez Performances et utilisation des ressources et Dimensionnement des clusters Amazon DocumentDB.

Il est important de dimensionner votre cluster Amazon DocumentDB avec les types d'instances appropriés pour des performances optimales. Pour plus d'informations, consultez les meilleures pratiques pourDimensionnement d'instance.

Vous pouvez utiliser le calculateur de dimensionnement Amazon DocumentDB* pour vous aider à estimer la taille de votre cluster Amazon DocumentDB.

* Non officiellement pris en charge parAWS.

Test de basculement

Vous souhaiterez peut-être observer comment votre application réagit au redémarrage d'un nœud principal Amazon DocumentDB, à un basculement du nœud principal ou à la suppression d'un nœud principal dans un cluster à nœuds multiples, ainsi que lorsque des nœuds répliques sont redémarrés ou supprimés. Cela vous aidera à confirmer que votre application est résiliente face à ces événements. Pour plus d'informations, consultez Test du basculement.

Pour comprendre les exceptions qu'une application doit tolérer et comment les gérer efficacement, consultez Création d'applications résilientes avec Amazon DocumentDB.

> **Note**
>
> Rien ne remplace le test de votre charge de travail sur Amazon DocumentDB

Étape 3 : migrer les données

Après une validation de principe réussie, migrez vos données vers Amazon DocumentDB. La plupart de nos clients utilisent des approches de migration en ligne ou hors ligne pour migrer leurs données.

Migration en ligne

À l'aide de la méthode de migration en ligne, vous pouvez migrer des données de votre base de données source, allant de quelques gigaoctets à plusieurs téraoctets, vers Amazon DocumentDB avec un temps d'arrêt quasi nul. Pour plus d'informations, consultez AWS Database Migration Service (AWS DMS).

Si vous migrez depuis une base de données MongoDB, vous pouvez l'AWS DMSutiliser pour effectuer un chargement complet et répliquer les modifications en cours.

Pour un step-by-step processus, consultez la section Migration vers Amazon DocumentDB avec la méthode en ligne.

Des informations supplémentaires sont disponibles dans la AWS Database Migration Service section Utiliser Amazon DocumentDB comme cible du Guide de l'AWS Database Migration Serviceutilisateur.

Points à noter concernant AWS DMS :

- Segmentation : lors de la migration de bases de données de plusieurs téraoctets à l'aide deAWS DMS, cela peut être lent avec les paramètres par défaut, car le chargement complet du DMS se fait par défaut sur un seul thread par collection, ce qui entraîne des temps de migration plus longs. Pour accélérer le chargement complet lors de migrations de bases de données volumineuses, vous pouvez utiliser la fonctionnalité de segmentation dansAWS DMS.

  Pour plus de détails sur l'utilisation de la segmentation avecAWS DMS, consultez la section Utilisation de la segmentation automatique avec AWS DMS.
- Type d'instance DMS : pour accélérer la migration des données, vous devez choisir la bonne instance DMS.

Migration hors ligne

La migration hors ligne est l'approche la plus simple pour déplacer des bases de données vers Amazon DocumentDB. Cette approche est principalement utilisée pour les POC et pour les charges de travail qui peuvent nécessiter des interruptions d'écriture pendant la migration.

Pour un step-by-step processus, consultez Migrer de MongoDB vers Amazon DocumentDB à l'aide de la méthode hors ligne.

Étape 4 : Validation des données

Une fois les données migrées avec succès, validez leur exactitude afin de gagner en confiance. Sur la console AWS DMS des tâches de migration, vous pouvez trouver les métriques relatives aux données migrées. Pour plus d'informations, consultez la section Vérifier les données migrées.

Vous pouvez également utiliser l' DataDiffer outil Amazon DocumentDB* pour valider la cohérence des données entre les collections source et cible.

* Non officiellement pris en charge parAWS.

Étape 5 : transfert de candidature

Cela implique de modifier la chaîne de connexion à la base de données de votre application pour utiliser votre cluster Amazon DocumentDB.

Pour plus d'informations sur la connexion à Amazon DocumentDB, consultez. Connexion à Amazon DocumentDB en tant qu'ensemble de réplicas

Migration en ligne

Une fois le chargement complet des données terminé, AWS DMS continue de répliquer les modifications en cours depuis votre source vers Amazon DocumentDB. Une fois les modifications prises en compte et les contrôles de validation des données terminés, vous pouvez effectuer un transfert vers Amazon DocumentDB.

Migration hors ligne

Une fois le chargement complet des données et les contrôles de validation des données terminés, vous pouvez effectuer le transfert vers Amazon DocumentDB.

Ressources supplémentaires

Voici quelques ressources supplémentaires qui pourraient vous aider dans votre migration :

- Vidéo : Bonnes pratiques pour la migration vers Amazon DocumentDB
- Vidéo : Commencer à utiliser l'observabilité et la surveillance d'Amazon DocumentDB
- Utilitaires supplémentaires : Amazon DocumentDB Tools *

- Guide du développeur de solutions de migration : Migration vers Amazon DocumentDB

* Non officiellement pris en charge parAWS.

Mise à niveau sur place de la version majeure d'Amazon DocumentDB

Amazon DocumentDB ne rend les nouvelles versions des moteurs de base de données généralement disponibles qu'après des tests approfondis. Vous pouvez choisir comment et quand mettre à niveau vos clusters Amazon DocumentDB vers la nouvelle version.

Amazon DocumentDB prend actuellement en charge trois versions principales : Amazon DocumentDB 3.6, 4.0 et 5.0. Vous pouvez effectuer une mise à niveau de version majeure (MVU) sur place de votre base de données tout en conservant les mêmes points de terminaison, le même stockage et les mêmes balises que les clusters et pouvez continuer à utiliser vos applications sans aucune modification. Cette fonctionnalité est disponible gratuitement dans toutes les régions où Amazon DocumentDB 5.0 est disponible.

> **Important**
>
> Vos clusters Amazon DocumentDB ne seront pas disponibles lors de la mise à niveau de la version majeure sur place et vos clusters subiront plusieurs redémarrages. Les interruptions de mise à niveau peuvent varier d'un cluster à l'autre en fonction du nombre de collections, d'index, de bases de données et d'instances. Nous vous recommandons d'effectuer la mise à niveau pendant votre période de maintenance ou pendant les heures de faible utilisation. Une fois votre cluster mis à niveau, vous ne pouvez pas le rétrograder vers la version précédente, mais vous pouvez choisir de restaurer votre instantané de pré-mise à niveau sur un nouveau cluster.

Rubriques

Conditions préalables et limitations

Voici les conditions préalables et les limites de la mise à niveau des versions majeures sur place que vous devrez peut-être comprendre et respecter avant d'effectuer la mise à niveau :

- Type d'instance — Amazon DocumentDB 4.0/5.0 ne prend pas en charge les instances r4.*. Pour procéder à une mise à niveau de version majeure sur place, remplacez les instances r4.* par des instances r5.*. Pour plus d'informations, consultez Modification d'une instance Amazon DocumentDB. Consultez les instances prises en Classes d'instances prises en région charge en fonction de la version du moteur Amazon DocumentDB.
- Correctifs du système d'exploitation d'instance : une mise à niveau de version majeure sur place nécessite le dernier correctif du système d'exploitation (OS) pour pouvoir être effectuée. Appliquez toutes les actions de maintenance du système d'exploitation en attente sur les instances avant de procéder à la mise à niveau sur place. Pour plus d'informations, consultez Utilisation des mises à jour du système d'exploitation.

  > **Note**
  >
  > Dans certains cas, si vous avez des correctifs de moteur au niveau du cluster en attente, les correctifs du système d'exploitation de l'instance ne sont pas visibles. Vous devrez peut-être appliquer des correctifs de moteur au niveau du cluster avant de procéder à l'application des correctifs du système d'exploitation de l'instance et, par la suite, à la mise à niveau de la version majeure sur place. veuillez consulter Exécution d'une mise à jour du correctif de la version du moteur d'un cluster.

- La mise à niveau des versions majeures sur place est disponible dans toutes les régions où Amazon DocumentDB 5.0 est disponible.
- La mise à niveau des versions majeures sur place n'est pas prise en charge avec Amazon DocumentDB 4.0 comme version cible.
- À partir d'Amazon DocumentDB 4.0, «. » dans les noms d'utilisateur n'est pas pris en charge. Si vous effectuez une mise à niveau d'Amazon DocumentDB 3.6 vers la version 5.0 et que votre nom d'utilisateur contient » . «, veuillez recréer votre nom d'utilisateur sans » . », avant de passer au MVU sur place.
- La mise à niveau des versions majeures sur place n'est actuellement pas prise en charge sur les clusters globaux et les clusters élastiques Amazon DocumentDB.

> **Note**
>
> Pour mettre à niveau vos clusters globaux, supprimez vos clusters secondaires du cluster global, convertissez le cluster principal en cluster régional, effectuez une mise à niveau de version majeure sur place sur le cluster régional (principal), puis recréez le cluster global en ajoutant des clusters secondaires portant le même nom afin de conserver les mêmes points de terminaison que précédemment. Notez que des frais d'E/S seront facturés pendant que votre cluster principal mis à niveau réplique les données vers les clusters secondaires que vous venez d'ajouter. Pour obtenir des instructions détaillées sur la façon de supprimer des clusters secondaires d'un cluster global avant de les supprimer, consultezSupprimer un cluster d'un cluster global Amazon DocumentDB.

- Si vous disposez d'un grand nombre d'index (> 10 000) et que vous travaillez sur une instance plus petite (par exemple, t3.medium), vous devez faire évoluer votre instance principale vers une instance plus grande (par exemple, au moins r5.xlarge) afin de réserver suffisamment de mémoire dans l'instance pour effectuer la mise à niveau de la version majeure sur place. Vous pouvez choisir de réduire la taille de l'instance une fois que la mise à niveau de la version majeure sur place est terminée. Consultez les tableaux ci-dessous pour connaître le nombre maximal d'index pris en charge par type d'instance pour une mise à niveau de version majeure sur place :

  Pour les instances optimisées pour la mémoire (db.r5.*) :

Instance	Nombre maximum d'index pris en charge pour le MVU sur place
db.r5.large	100 000
db.r5.xlarge	200 000
db.r5.2xlarge	300 000
db.r5.4xlarge	400 000
db.r5.8xlarge	500 000
db.r5.12xlarge	700 000
db.r5.16xlarge	800 000

Instance	Nombre maximum d'index pris en charge pour le MVU sur place
db.r5.24xlarge	1 M

Pour les instances de performance éclatantes (db.t3, db.t4g)

Instance	Nombre maximum d'index pris en charge pour le MVU sur place
db.t4g.medium	3 KM
db.t3.medium	10 000

Pour les instances de graviton optimisées pour la mémoire (db.r6g.*) :

Instance	Nombre maximum d'index pris en charge pour le MVU sur place
db.r6g.large	100 000
db.r6g.xlarge	200 000
db.r6g.2xlarge	300 000
db.r6g.4xlarge	400 000
db.r6g.8xlarge	500 000
db.r6g.12xlarge	700 000
db.r6g.16xlarge	800 000

> **Note**
>
> Si vous avez plus d'un million d'index, contactez le AWS support et ne procédez pas à une mise à niveau de version majeure sur place.

Bonnes pratiques pour les mises à niveau des versions majeures sur place

Testez sur place les mises à niveau des versions majeures à l'aide de clusters clonés

1. Pour tester les mises à niveau des versions majeures sur place, nous vous recommandons d'utiliser la fonction de clonage rapide pour créer un clone de votre cluster cible. Aucun coût de stockage n'est nécessaire pour tester la mise à niveau d'une version majeure sur place sur un volume cloné, sauf si vous modifiez les données du cluster. Pour plus d'informations sur le clonage de volume, consultezClonage d'un volume pour un cluster Amazon DocumentDB.
2. Pour obtenir une estimation plus réaliste du temps nécessaire pour terminer la mise à niveau de la version majeure sur place, faites correspondre le nombre d'instances du cluster cloné au cluster cible.
3. Nous vous recommandons de tester entièrement le cluster Amazon DocumentDB 5.0 récemment mis à niveau pour détecter toute différence fonctionnelle afin de vous assurer que tout fonctionne comme prévu.

Avant une mise à niveau sur place d'une version majeure

1. Préparez un groupe de paramètres de cluster compatible avec les versions.

   Utilisez le groupe de paramètres de cluster par défaut d'Amazon DocumentDB pour la nouvelle version du moteur ou créez votre propre groupe de paramètres de cluster personnalisé pour la nouvelle version du moteur.

Si vous associez un groupe de paramètres de cluster Amazon DocumentDB dans le cadre de la demande de mise à niveau, la mise à niveau de la version majeure sur place redémarrera automatiquement le cluster pour appliquer le nouveau groupe de paramètres.

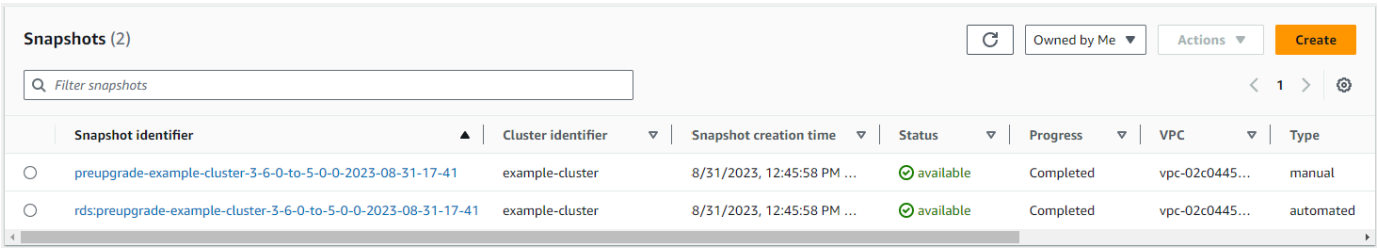
2. Assurez-vous que vous avez satisfait aux conditions requises pour une mise à niveau de version majeure sur place, comme indiqué dans la section Conditions préalables et limites.

3. Créez un instantané manuel.

   Le processus de mise à niveau crée un instantané de votre cluster de base de données lors de la mise à niveau. Il est vivement recommandé de créer votre propre instantané manuel avant le processus de mise à niveau. veuillez consulter Création d'un instantané manuel d'un cluster.

   > **Note**
   >
   > L'instantané automatique créé par le processus de mise à niveau ne sera pas automatiquement supprimé une fois la mise à niveau de la version majeure sur place terminée. Cet instantané n'entraînera aucun frais tant qu'il est conservé pendant la période de conservation. Vous pouvez choisir de supprimer cet instantané une fois que vous avez vérifié la réussite de la mise à niveau de votre cluster.

   Le cliché est nommé comme`preupgrade-<name>-<version>-<timestamp>`.



4. Vérifiez si vous avez déjà planifié une mise à niveau de la version majeure sur place de votre cluster.

   Si vous avez modifié le cluster et choisi de l'appliquer dans la fenêtre de maintenance suivante, le calendrier de mise à niveau des versions majeures sur place ne sera pas visible sur la console, mais vous pouvez le consulter dans la CLI. Vous pouvez exécuter la commande suivante pour vérifier si une mise à niveau de version majeure sur place est déjà planifiée :

```
aws docdb describe-db-cluster \
--region $REGION \
```

```
--db-cluster-identifier $CLUSTER_NAME

"PendingModifiedValues": {
    "EngineVersion": "5.0.0"
},
```

5. Effectuez plusieurs essais à l'aide d'un clone de volume dans des environnements inférieurs pour tester le cluster après la mise à niveau de la version majeure sur place, quel que soit le plan d'exécution et les différences fonctionnelles. Nous recommandons le clonage avec le même nombre et la même taille d'instances afin d'obtenir une meilleure estimation du temps d'exécution de la mise à niveau des versions majeures sur place. Pour plus d'informations, consultez Clonage d'un volume pour un cluster Amazon DocumentDB.
6. Si l'étape précédente est réussie, procédez à la mise à niveau de la version majeure sur place sur le cluster de production.

Lors d'une mise à niveau d'une version majeure sur place

Vous pouvez suivre la progression de la mise à niveau de votre version majeure sur place en vous abonnant aux événements de maintenance du cluster. Une fois la mise à niveau terminée, vous recevrez l'événement « La version majeure du cluster de base de données a été mise à niveau ». Cet événement, ainsi que d'autres événements survenant pendant la mise à niveau, apparaissent dans la section « Événements et balises » de la page détaillée du cluster dans la console Amazon DocumentDB. Le statut du cluster passe ensuite de « mise à niveau » à « disponible ».

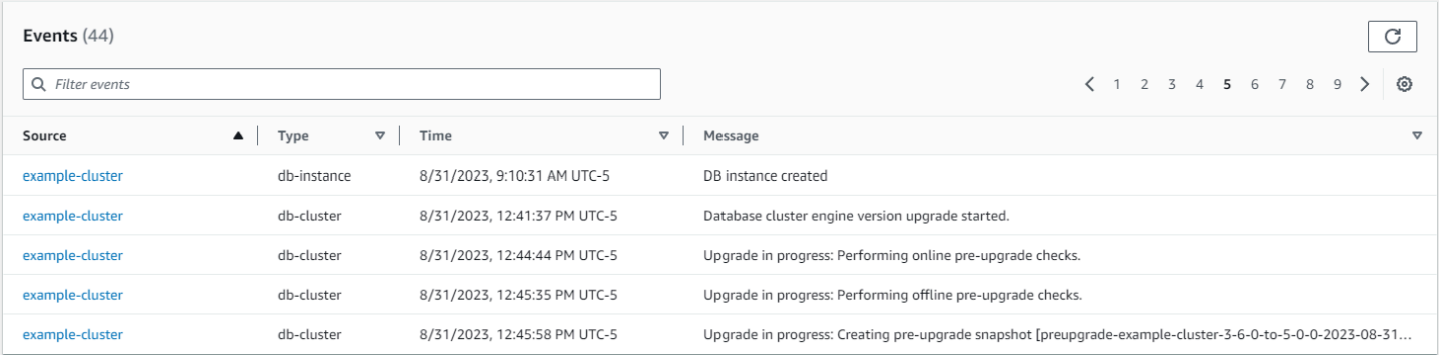
À partir de la CLI, vous pouvez exécuter `aws docdb create-event-subscription` pour créer des événements et `aws docdb describe-events` suivre les progrès. Vous pouvez également configurer des notifications d'événements pour les événements ci-dessus à Amazon SNS en tant que cible pour les notifications par e-mail, messages push et autres méthodes. Pour plus d'informations, consultez Abonnement aux abonnements aux événements Amazon DocumentDB.

La mise à niveau de la version majeure sur place génère les événements suivants au cours de la mise à niveau :

- <cluster-name><timestamp>Mise à niveau en cours : création d'un instantané préalable à la mise à niveau [preupgrade- -]
- Mise à niveau en cours : volume de clonage.
- Mise à niveau en cours : mise à niveau de Writer.
- Mise à niveau en cours : mise à niveau des lecteurs.

- La version majeure du cluster de base de données a été mise à niveau.

Les événements sont également visibles sur la console, sous la page Événements :



Dans le AWS CLI, vous pouvez utiliser les commandes suivantes pour suivre les progrès :

```
aws docdb describe-events --source-identifier $CLUSTER_NAME --source-type db-cluster
{
    "Events": [
        {
            "SourceIdentifier": "mycluster",
            "SourceType": "db-cluster",
            "Message": "Database cluster engine version upgrade started.",
            "EventCategories": [
                "maintenance"
            ],
            "Date": "2023-07-11T23:20:32.444000+00:00",
            "SourceArn": "arn:aws:rds:us-east-1:xxxx:cluster:mycluster"
        }
    ]
}
```

Après une mise à niveau de la version majeure sur place

Pour Amazon DocumentDB 3.6, ajoutez une balise au cluster pour indiquer que le cluster a été mis à niveau vers Amazon DocumentDB 5.0 à partir d'Amazon DocumentDB 3.6 par opposition à un cluster Amazon DocumentDB 5.0 récemment créé. Reportez-vous à la section sur les différences entre un cluster Amazon DocumentDB 5.0 mis à niveau et un nouveau cluster Amazon DocumentDB 5.0.

Prenez un instantané manuel une fois la mise à niveau de la version majeure sur place terminée, au cas où vous auriez besoin de rétablir l'état après la mise à niveau. Le processus de capture automatique reprendra dès que la mise à niveau de la version majeure sur place sera terminée.

L'instantané manuel n'entraînera aucun frais tant qu'il est conservé pendant la période de conservation.

Pour utiliser les nouvelles fonctionnalités associées à Amazon DocumentDB 5.0, par exemple le chiffrement au niveau des champs côté client, nous vous recommandons de mettre à niveau la version de votre pilote vers la version de l'API MongoDB 5.0. Pour plus d'informations, consultez Nouveautés d'Amazon DocumentDB 5.0 la liste des fonctionnalités d'Amazon DocumentDB 5.0.

Important

Immédiatement après avoir effectué la mise à niveau sur place de la version majeure (MVU), votre cluster Amazon DocumentDB 5.0 reremplit les métadonnées de l'index, sur la base desquelles le moteur de base de données optimise les plans d'exécution des requêtes. Les performances de requête attendues sur votre cluster Amazon DocumentDB reprendront une fois le processus de recalcul des métadonnées d'index terminé. Ce processus prend généralement quelques minutes, mais peut durer jusqu'à deux heures selon le nombre d'index de votre cluster.
En outre, un redémarrage immédiat, un basculement ou une augmentation ou une réduction de la taille de votre instance Writer après la mise en place d'un MVU peuvent perturber le processus de calcul des métadonnées d'index sur votre cluster. Une fois le MVU sur place terminé, nous vous recommandons d'apporter ces modifications une fois que vous aurez observé les performances de requête attendues sur votre cluster Amazon DocumentDB 5.0. Veuillez contacter le AWS support si vous constatez que cette baisse de performance temporaire persiste pendant plus de deux heures après la mise en place du MVU.

Testez entièrement le cluster Amazon DocumentDB 5.0 mis à niveau pour vous assurer que tout fonctionne comme prévu.

Note

Après avoir effectué un MVU sur place sur un cluster Amazon DocumentDB avec les flux de modifications activés, les événements du flux de modifications précédents sont préservés et peuvent être repris à l'aide de ou. `resumeToken` `startAtOperationTime` Comme c'est le cas dans tout cluster Amazon DocumentDB nouvellement créé, les journaux d'événements des flux de modifications antérieurs à cette valeur

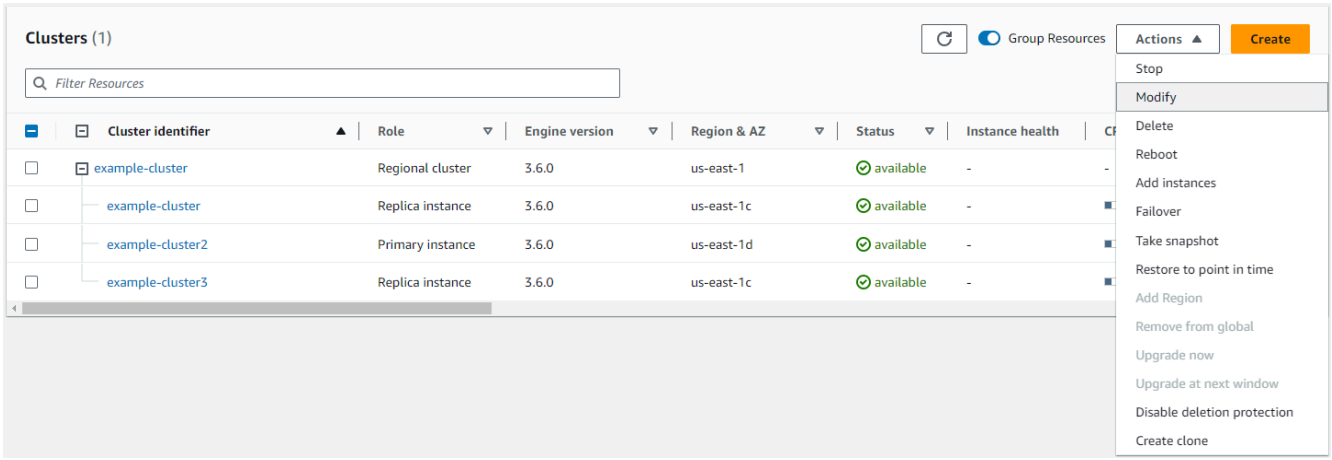
`change_stream_log_retention_duration` sont supprimés si leur taille est supérieure à 51 200 Mo.

Réalisation d'une mise à niveau de version majeure sur place

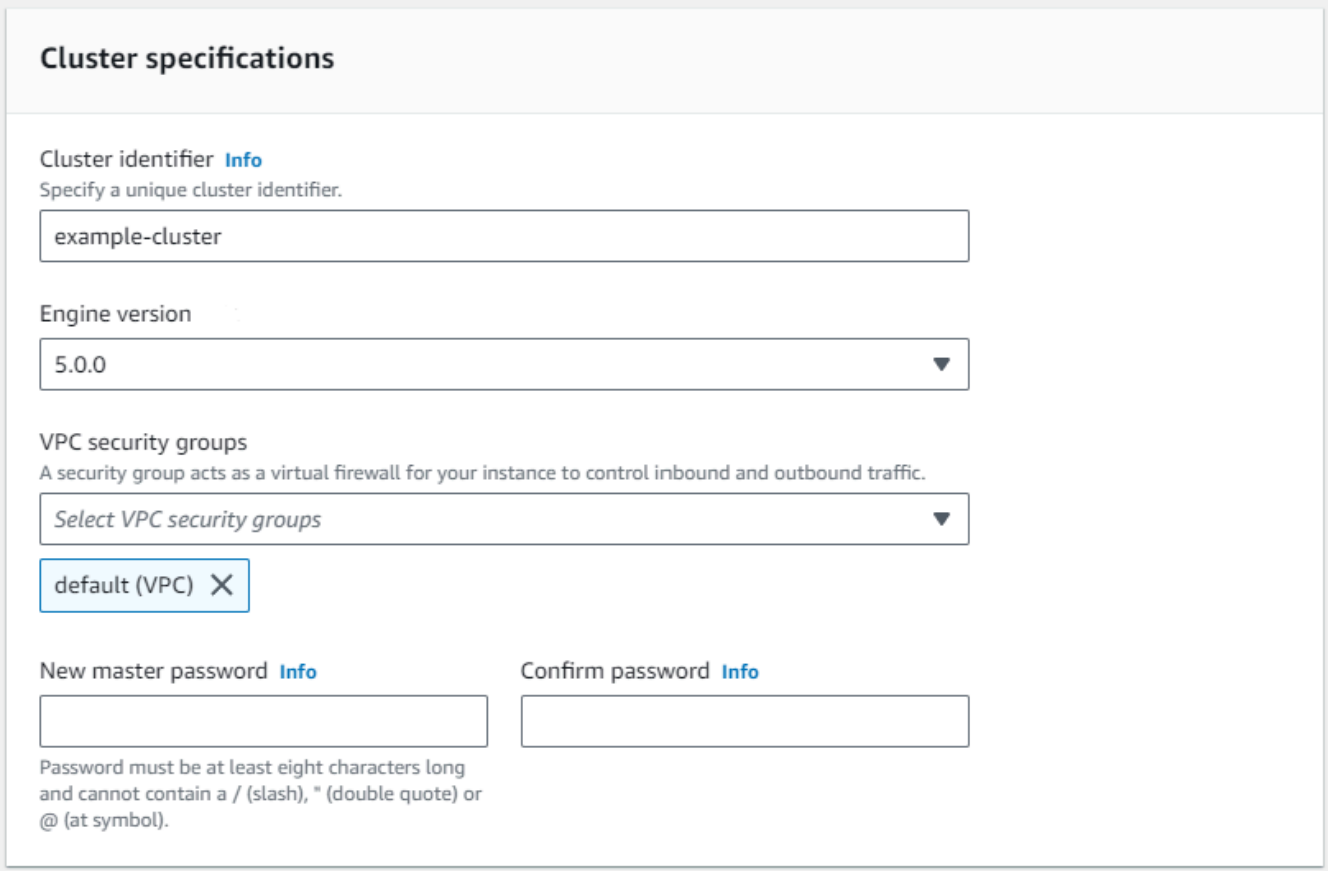
Using the AWS Management Console

Pour effectuer une mise à niveau de version majeure sur place à l'aide de AWS Management Console :

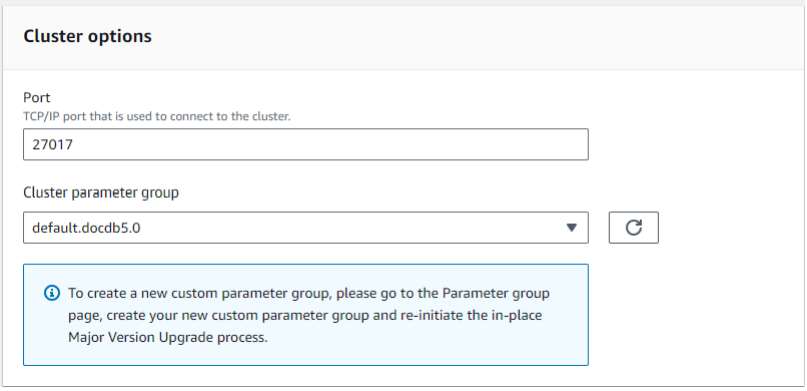
1. Connectez-vous à la console Amazon DocumentDB AWS Management Consoleet ouvrez-la.
2. Dans le tableau Clusters, sélectionnez le cluster source, cliquez sur Actions, puis sur Modifier.



3. Dans la boîte de dialogue Modifier le cluster de la section Spécifications du cluster, choisissez la version de base de données ciblée (5.0) dans le menu déroulant Version du moteur.

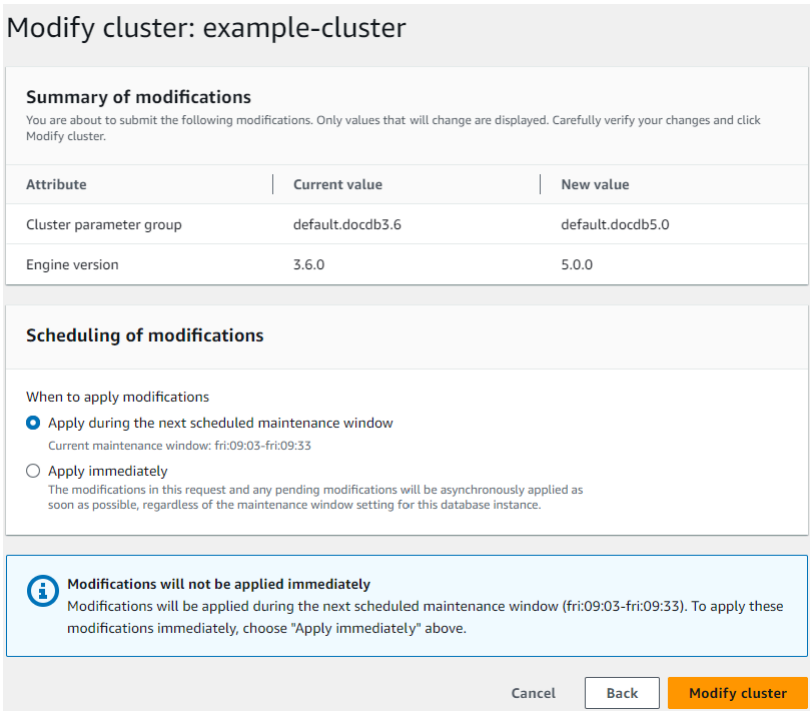


4. Dans la section Options du cluster, choisissez le groupe de paramètres de cluster approprié (default.docdb5.0) ou un groupe de paramètres créé sur mesure.

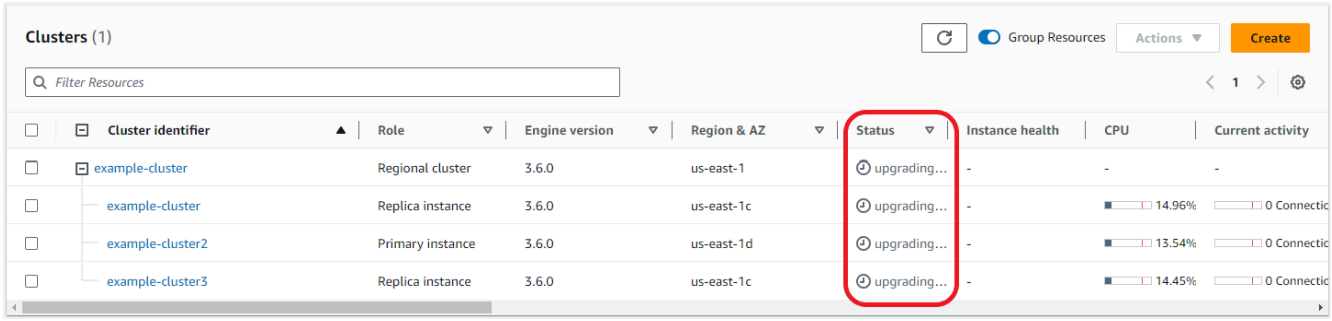


5. Une fois terminé, faites défiler l'écran vers le bas et choisissez Continuer.
6. Dans la section Planification des modifications, choisissez votre plan de planification préféré : appliquez-le immédiatement ou appliquez-le dans la fenêtre de maintenance suivante.

   Ensuite, choisissez Modify cluster (Modifier le cluster).



7. Dans le tableau des clusters, notez l'état de votre cluster lors de sa mise à niveau :



Using the AWS CLI

Utilisez l'`modify-db-cluster`API avec la version du moteur et le jeu d'`allow-major-version-upgrade`indicateurs souhaités :

```
aws docdb modify-db-cluster \
 --db-cluster-identifier $CLUSTER_NAME \
 --allow-major-version-upgrade \
 --engine-version 5.0 \
 --apply-immediately \
 --cluster-parameter-group $PARAMETER_GROUP \
 --region $REGION
```

Différences entre les clusters mis à niveau Amazon DocumentDB 3.6/4.0 à 5.0 et les nouveaux clusters Amazon DocumentDB 5.0

- Comparaisons de sous-documents pour plusieurs types de données numériques :
  - Si le cluster est migré depuis Amazon DocumentDB 3.6, il héritera du comportement de comparaison des sous-documents Amazon DocumentDB 3.6. La différence fonctionnelle est limitée aux types numériques (tels que Long, Double, Decimal128) dans un sous-document. Par exemple, `{a: {b: {NumberLong(1)}}` ce n'est pas égal `{a: {b: 1}}` dans Amazon DocumentDB 3.6, alors qu'ils sont comparés comme égaux dans Amazon DocumentDB 4.0 et versions ultérieures.
  - Ce comportement de comparaison de sous-documents n'existe que dans Amazon DocumentDB 3.6 et dans les clusters Amazon DocumentDB 5.0 qui ont été mis à niveau à partir de la version 3.6 à l'aide d'une mise à niveau de version majeure sur place. Cela ne s'applique pas aux clusters Amazon DocumentDB 5.0 nouvellement créés.
- Une mise à niveau de version majeure sur place conserve les index d'origine du cluster mis à niveau. En règle générale, nous vous recommandons de supprimer et de recréer vos index une fois le MVU en place terminé avec succès. Avec Amazon DocumentDB 5.0, nous avons amélioré l'efficacité globale du processus de collecte des déchets, en particulier pour les faibles indices de cardinalité. Si vous avez déjà rencontré des problèmes liés à la collecte des déchets sur vos clusters Amazon DocumentDB 3.6 ou 4.0, ces clusters bénéficieront de la suppression et de la recréation des index après le MVU. Il n'est pas obligatoire de recréer des index. Cependant, la recréation d'un index peut impliquer des E/S et du temps supplémentaires. Pour plus d'informations, consultez Gestion des index Amazon DocumentDB.

> **Note**
>
> Pour obtenir la liste des différences fonctionnelles entre Amazon DocumentDB 3.6/4.0 et Amazon DocumentDB 5.0, consultez. Compatibilité avec MongoDB

Résolution des problèmes liés à une mise à niveau d'une version majeure sur place

- En cas d'échec, la mise à niveau de la version majeure sur place tentera d'annuler la mise à niveau afin de rétablir le dernier état opérationnel du cluster avant le début de la mise à niveau. Une restauration réussie générera un événement : « Le cluster de base de données est dans un état qui ne peut pas être mis à niveau : le cluster DocumentDB est dans un état dans lequel la mise à niveau de la version majeure ne peut pas être effectuée correctement. » À ce stade, vous devez contacter l'équipe de AWS support pour résoudre les problèmes et réessayer la mise à niveau de la version. Vous pouvez continuer à utiliser votre charge de travail comme avant. Dans tous les autres rares scénarios où la mise à niveau prend plus de temps que prévu, veuillez contacter l'équipe d' AWS assistance pour obtenir de l'aide.
- Une fois que votre MVU sur place est terminé avec succès, votre cluster mis à niveau peut subir une dégradation temporaire des performances et une utilisation élevée du processeur pendant une courte période, pendant que le processus d'actualisation des métadonnées d'index est en cours d'exécution. Si la dégradation des performances persiste pendant plus de 2 heures, contactez le AWS support.

Sécurité dans Amazon DocumentDB

Chez AWS, la sécurité dans le cloud est notre priorité numéro 1. En tant que client AWS, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des organisations les plus pointilleuses en termes de sécurité.

La sécurité est une responsabilité partagée entre AWS et vous-même. Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lorsque vous utiliser Amazon DocumentDB. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est responsable de la protection de l'infrastructure qui exécute des services AWS dans le Cloud AWS. AWS vous fournit également les 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 conformité AWS. Pour en savoir plus sur les programmes de conformité qui s'appliquent à Amazon DocumentDB (compatible avec MongoDB), veuillez consulter AWSServices concernés par le programme de conformité.
- Sécurité dans le cloud : votre responsabilité est déterminée par le service AWS que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre organisation, et la législation et la réglementation applicables.

> **Note**
>
> Ce chapitre s'applique à la fois aux clusters basés sur des instances et aux clusters élastiques. Pour plus d'informations, consultez les rubriques ci-dessous.

Vous pouvez également apprendre à utiliser d'autresAWS services qui vous permettent de surveiller et de sécuriser vos ressources Amazon DocumentDB. Les rubriques suivantes vous montrent comment configurer Amazon DocumentDB pour répondre à vos objectifs de sécurité et de conformité.

Rubriques

- Protection des données dans Amazon DocumentDB
- Identity and Access Management pour Amazon DocumentDB
- Gestion des utilisateurs Amazon DocumentDB

- Accès à la base de données à l'aide du contrôle d'accès basé sur les rôles
- Journalisation et surveillance dans Amazon DocumentDB
- Mise à jour de vos certificats TLS Amazon DocumentDB
- Mise à jour de vos certificats TLS Amazon DocumentDB — (USA Ouest) GovCloud
- Validation de conformité dans Amazon DocumentDB
- Résilience dans Amazon DocumentDB
- Sécurité de l'infrastructure dans Amazon DocumentDB
- Bonnes pratiques de sécurité pour Amazon DocumentDB
- Audit des événements Amazon DocumentDB

Protection des données dans Amazon DocumentDB

Le modèle de responsabilité AWS partagée de s'applique à la protection des données dans. Comme décrit dans ce modèle, AWS est responsable de la protection de l'infrastructure globale sur laquelle l'ensemble du AWS Cloud s'exécute. 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 en savoir plus 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 AWSet RGPD (Règlement général sur la protection des données) sur le AWSBlog de sécurité.

À des fins de protection des données, nous vous recommandons de protéger les informations d'identification Compte AWS et de configurer les comptes utilisateur individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (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.
- Utilisez les certificats SSL/TLS pour communiquer avec les ressources AWS. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez une API (Interface de programmation) et le journal de l'activité des utilisateurs avec AWS CloudTrail.
- Utilisez des solutions de chiffrement AWS, ainsi que tous les contrôles de sécurité par défaut au sein des 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 FIPS (Federal Information Processing Standard) 140-2 lorsque vous accédez à AWS via une CLI (Interface de ligne de commande) ou une API (Interface de programmation), utilisez un point de terminaison FIPS (Federal Information Processing Standard). Pour en savoir plus sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez Federal Information Processing Standard (FIPS) 140-2 (Normes de traitement de l'information fédérale).

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 Name (Nom). Cela inclut lorsque vous travaillez avec Amazon DocumentDB ou un autre outil à Services AWS l'aide de la console, de l'API ou AWS des AWS CLI SDK. Toutes les données que vous saisissez 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.

Rubriques

Chiffrement côté client au niveau du champ

Le chiffrement au niveau du champ (FLE) côté client Amazon DocumentDB vous permet de crypter les données sensibles de vos applications clientes avant leur transfert vers un cluster Amazon DocumentDB. Les données sensibles restent chiffrées lorsqu'elles sont stockées et traitées dans un cluster et sont déchiffrées dans l'application cliente lors de leur récupération.

Rubriques

- Limites

Démarrer

La configuration initiale du fichier FLE côté client dans Amazon DocumentDB est un processus en quatre étapes qui comprend la création d'une clé de chiffrement, l'association d'un rôle à l'application, la configuration de l'application et la définition du fonctionnement du CRUD à l'aide d'options de chiffrement.

Rubriques

- Étape 1 : créer les clés de chiffrement
- Étape 2 : associer un rôle à l'application
- Étape 3 : Configurer l'application
- Étape 4 : définir une opération CRUD
- Exemple : fichier de configuration du chiffrement au niveau du champ côté client

Étape 1 : créer les clés de chiffrement

À l'aide deAWS Key Management Service, créez une clé symétrique utilisée pour crypter et déchiffrer le champ de données sensibles et fournissez-lui les autorisations d'utilisation IAM nécessaires. AWS KMSstocke la clé client (CK) qui est utilisée pour crypter les clés de données (DKs). Nous vous recommandons de stocker la clé client dans KMS pour renforcer votre sécurité. La clé de données est la clé secondaire qui est stockée dans une collection Amazon DocumentDB et qui est requise pour crypter les champs sensibles avant de stocker le document dans Amazon DocumentDB. La clé client chiffre la clé de données qui, à son tour, chiffre et déchiffre vos données. Si vous utilisez un cluster global, vous pouvez créer une clé multirégion qui peut être utilisée par différents rôles de service dans différentes régions.

Pour plus d'informations surAWS Key Management Service, notamment sur la création d'une clé, consultez le Guide du développeur du service de gestion desAWS clés.

Étape 2 : associer un rôle à l'application

créer une politique IAM avec desAWS KMS autorisations appropriées. Cette politique autorise les identités IAM auxquelles elle est attachée à chiffrer et déchiffrer la clé KMS spécifiée dans le champ de ressource. Votre application assume ce rôle IAM pour s'authentifierAWS KMS.

La politique doit être similaire à ceci :

```
{ "Effect": "Allow",
"Action": ["kms:Decrypt", "kms:Encrypt"],
"Resource": "Customer Key ARN"
}
```

Étape 3 : Configurer l'application

Vous avez désormais défini une clé client dansAWS KMS, créé un rôle IAM et lui avez fourni les autorisations IAM appropriées pour accéder à la clé client. Importez les packages obligatoires.

```
import boto3
import json
import base64
from pymongo import MongoClient
from pymongo.encryption import (Algorithm,
                                ClientEncryption)
```

```
# create a session object:
my_session = boto3.session.Session()

# get access_key and secret_key programmatically using get_frozen_credentials() method:
 current_credentials = my_session.get_credentials().get_frozen_credentials()
```

1. Spécifiez « aws » comme type de fournisseur KMS et saisissez les informations d'identification de votre compte qui ont été récupérées à l'étape précédente.

   ```
   provider = "aws"
   kms_providers = {
       provider: {
           "accessKeyId": current_credentials.access_key,
           "secretAccessKey": current_credentials.secret_key
       }
   }
   ```

2. Spécifiez la clé client utilisée pour chiffrer la clé de données :

   ```
   customer_key = {
   “region”: “AWS region of the customer_key”,
       “key”: “customer_key ARN”
   ```

```
}

key_vault_namespace = "encryption.dataKeys"

key_alt_name = 'TEST_DATA_KEY'
```

3. Configurez MongoClient l'objet :

```
client = MongoClient(connection_string)

coll = client.test.coll
coll.drop()

client_encryption = ClientEncryption(
    kms_providers, # pass in the kms_providers variable from the previous step
    key_vault_namespace = key_vault_namespace,
    client,
    coll.codec_options
)
```

4. Générez votre clé de données :

```
data_key_id = client_encryption.create_data_key(provider,
    customer_key,
    key_alt_name = [key_alt_name])
```

5. Récupérez votre clé de données existante :

```
data_key = DataKey("aws",
    master_key = customer_key)
key_id = data_key["_id"]
data_key_id = client[key_vault_namespace].find_one({"_id": key_id})
```

Étape 4 : définir une opération CRUD

Définissez l'opération CRUD à l'aide des options de chiffrement.

1. Définissez la collection pour écrire/lire/supprimer un seul document :

```
coll = client.gameinfo.users
```

2. Chiffrement explicite : cryptez les champs et insérez :

> **Note**
>
> Exactement l'un des termes « key_id » ou « key_alt_name » doit être fourni.

```
encrypted_first_name = client_encryption.encrypt(
    "Jane",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
    key_alt_name=data_key_id
)
encrypted_last_name = client_encryption.encrypt(
    "Doe",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
    key_alt_name=data_key_id
)
encrypted_dob = client_encryption.encrypt(
    "1990-01-01",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Random,
    key_alt_name=data_key_id
)

coll.insert_one(
    {"gamerTag": "jane_doe90",
    "firstName": encrypted_first_name,
    "lastName": encrypted_last_name,
    "dateOfBirth":encrypted_dob,
    "Favorite_games":["Halo","Age of Empires 2","Medal of Honor"]
})
```

Exemple : fichier de configuration du chiffrement au niveau du champ côté client

Dans les exemples suivants, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations.

```
# import python packages:
import boto3
import json
import base64
from pymongo import MongoClient
from pymongo.encryption import (Algorithm,
```

```
                                ClientEncryption)

def main():

    # create a session object:
    my_session = boto3.session.Session()

    # get aws_region from session object:
    aws_region = my_session.region_name

    # get access_key and secret_key programmatically using get_frozen_credentials()
 method:
    current_credentials = my_session.get_credentials().get_frozen_credentials()
    provider = "aws"

    # define the kms_providers which is later used to create the Data Key:
    kms_providers = {
        provider: {
            "accessKeyId": current_credentials.access_key,
            "secretAccessKey": current_credentials.secret_key
        }
    }

    # enter the kms key ARN. Replace the example ARN value.
    kms_arn = "arn:aws:kms:us-east-1:123456789:key/abcd-efgh-ijkl-mnop"
    customer_key = {
        "region": aws_region,
        "key":kms_arn
    }

    # secrets manager is used to strore and retrieve user credentials for connecting to
 an Amazon DocumentDB cluster.
    # retrieve the secret using the secret name. Replace the example secret key.
    secret_name = "/dev/secretKey"
    docdb_credentials = json.loads(my_session.client(service_name = 'secretsmanager',
 region_name = "us-east-1").get_secret_value(SecretId = secret_name)['SecretString'])

    connection_params = '/?tls=true&tlsCAFile=global-
bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false'
    conn_str = 'mongodb://' + docdb_credentials["username"] + ':' +
 docdb_credentials["password"] + '@' + docdb_credentials["host"] + ':' +
 str(docdb_credentials["port"]) + connection_params
    client = MongoClient(conn_str)
```

```
    coll = client.test.coll
    coll.drop()

    # store the encryption data keys in a key vault collection (having naming
convention as db.collection):
    key_vault_namespace = "encryption.dataKeys"
    key_vault_db_name, key_vault_coll_name = key_vault_namespace.split(".", 1)

    # set up the key vault (key_vault_namespace) for this example:
    key_vault = client[key_vault_db_name][key_vault_coll_name]
    key_vault.drop()
    key_vault.create_index("keyAltNames", unique=True)

    client_encryption = ClientEncryption(
        kms_providers,
        key_vault_namespace,
        client,
        coll.codec_options)

    # create a new data key for the encrypted field:
    data_key_id = client_encryption.create_data_key(provider, master_key=customer_key,
key_alt_names=["some_key_alt_name"], key_material = None)

    # explicitly encrypt a field:
    encrypted_first_name = client_encryption.encrypt(
    "Jane",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
    key_id=data_key_id
    )
    coll.insert_one(
    {"gamerTag": "jane_doe90",
    "firstName": encrypted_first_name
    })
    doc = coll.find_one()
    print('Encrypted document: %s' % (doc,))

    # explicitly decrypt the field:
    doc["encryptedField"] = client_encryption.decrypt(doc["encryptedField"])
    print('Decrypted document: %s' % (doc,))

    # cleanup resources:
    client_encryption.close()
    client.close()
```

```
    if __name__ == "__main__":
        main()
```

Recherche dans un fichier FLE côté client

Amazon DocumentDB prend en charge les requêtes d'égalité de points avec un FLE côté client. Les requêtes d'inégalité et de comparaison peuvent renvoyer des résultats inexacts. Les opérations de lecture et d'écriture peuvent avoir un comportement inattendu ou incorrect par rapport à l'exécution de la même opération sur la valeur déchiffrée.

Par exemple, pour rechercher des filtres pour des documents dont le score de joueur est supérieur à 500 :

```
db.users.find( {
    "gamerscore" : { $gt : 500 }
})
```

Le client utilise une méthode de cryptage explicite pour crypter la valeur de la requête :

```
encrypted_gamerscore_filter = client_encryption.encrypt(
    500,
        Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
        key_alt_name=data_key_id
        )

db.users.find( {
    "gamerscore" : { $gt : encrypted_gamerscore_filter }
} )
```

Lors de l'opération de recherche, Amazon DocumentDB compare la valeur chiffrée de 500 aux valeurs des champs cryptés stockées dans chaque document à l'aide du contrôle des inégalités supérieures à. La vérification des inégalités dans l'opération de recherche peut renvoyer un résultat différent lorsqu'elle est effectuée à l'aide de données et de valeurs déchiffrées, même si l'opération réussit à générer des résultats.

Limites

Les limites suivantes s'appliquent au chiffrement au niveau du champ côté client Amazon DocumentDB :

- Amazon DocumentDB prend uniquement en charge les requêtes d'égalité de points. Les requêtes d'inégalité et de comparaison peuvent renvoyer des résultats inexacts. Les opérations de lecture et d'écriture peuvent avoir un comportement inattendu ou incorrect par rapport à l'exécution de la même opération sur la valeur déchiffrée. Pour rechercher des filtres pour les documents dont le score de joueur est supérieur à 500.

```
db.users.find( {
    "gamerscore" : { $gt : 500 }
    })
```

Le client utilise une méthode de cryptage explicite pour crypter la valeur de la requête.

```
encrypted_gamerscore_filter = client_encryption.encrypt(
    500,
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
    key_alt_name=data_key_id
)

db.users.find({
    "gamerscore" : { $gt : encrypted_gamerscore_filter }
})
```

Lors de l'opération de recherche, Amazon DocumentDB compare la valeur chiffrée de 500 aux valeurs des champs cryptés stockées dans chaque document à l'aide du contrôle des inégalités supérieures à. La vérification des inégalités dans l'opération de recherche peut renvoyer un résultat différent lorsqu'elle est effectuée à l'aide de données et de valeurs déchiffrées, même si l'opération réussit à générer des résultats.

- Amazon DocumentDB ne prend pas en charge les fichiers FLE côté client explicites provenant du Mongo Shell. Toutefois, cette fonctionnalité fonctionne avec tous les pilotes pris en charge.

Chiffrement des données Amazon DocumentDB au repos

> **Note**
>
> AWS KMS remplace le terme clé principale client (CMK) par AWS KMS key et clé KMS. Le concept n'a pas changé. Pour éviter les changements de rupture, AWS KMS conserve quelques variations de ce terme.

Vous chiffrez les données au repos dans votre cluster Amazon DocumentDB en spécifiant l'option de chiffrement du stockage lorsque vous créez votre cluster. Le chiffrement du stockage est activé au niveau du cluster et appliqué à toutes les instances, y compris l'instance principale et toutes les réplicas. Elle est également appliquée au volume de stockage, aux données, aux index, aux journaux, aux sauvegardes automatisées et aux instantanés de votre cluster.

Amazon DocumentDB utilise la norme de chiffrement avancée 256 bits (AES-256) pour chiffrer vos données à l'aide des clés de chiffrement stockées dansAWS Key Management Service (AWS KMS). Lorsque vous utilisez un cluster Amazon DocumentDB avec le chiffrement au repos activé, vous n'avez pas besoin de modifier la logique de votre application ou la connexion client. Amazon DocumentDB gère le chiffrement et le déchiffrement de vos données de façon transparente, avec un impact minimal sur les performances.

Amazon DocumentDB intègreAWS KMS et utilise une méthode connue sous le nom de chiffrement des enveloppes pour protéger vos données. Lorsqu'un cluster Amazon DocumentDB est chiffréAWS KMS à l'aide d'unAWS KMS, Amazon DocumentDB vous demande d'utiliser votre clé KMS pour générer une clé de données chiffrée afin de chiffrer le volume de stockage. La clé de données chiffrée est chiffrée à l'aide de la clé KMS que vous définissez et est stockée avec les données cryptées et les métadonnées de stockage. Lorsqu'Amazon DocumentDB a besoin d'accéder à vos données chiffrées, il demande de déchiffrer la clé de données chiffréeAWS KMS à l'aide de votre clé KMS et met en cache la clé de données en clair en mémoire afin de crypter et de déchiffrer efficacement les données du volume de stockage.

La fonctionnalité de chiffrement du stockage d'Amazon DocumentDB est disponible pour toutes les tailles d'instance prises en charge et partoutRégions AWS où Amazon DocumentDB est disponible.

Activation du chiffrement au repos pour un cluster Amazon DocumentDB

Vous pouvez activer ou désactiver le chiffrement au repos sur un cluster Amazon DocumentDB lorsque le cluster est provisionné à l'aide duAWS Management Console ou duAWS Command Line Interface (AWS CLI). Le chiffrement au repos est activé par défaut pour les clusters que vous créez à l'aide de la console. Le chiffrement au repos est désactivé par défaut pour les clusters que vous créez à l'aide de l’AWS CLI. Par conséquent, vous devez activer explicitement le chiffrement au repos à l'aide du paramètre `--storage-encrypted`. Dans les deux cas, une fois le cluster créé, vous ne pouvez pas modifier l'option de chiffrement au repos.

Amazon DocumentDB permetAWS KMS de récupérer et de gérer les clés de chiffrement et de définir les politiques qui contrôlent la manière dont ces clés peuvent être utilisées. Si vous ne spécifiez aucun identifiant deAWS KMS clé, Amazon DocumentDB utilise la clé KMS du serviceAWS géré

par défaut. Amazon DocumentDB crée une clé KMS distincte pour chacunRégion AWS de vos fichiersCompte AWS. Pour plus d'informations, consultez Concepts AWS Key Management Service.

Pour commencer à créer votre propre clé KMS, consultez la section Mise en route du Guide duAWS Key Management Service développeur.

> **Important**
>
> Vous devez utiliser une clé KMS de chiffrement symétrique pour chiffrer votre cluster, Amazon DocumentDB ne prend en charge que les clés KMS de chiffrement symétrique. N'utilisez pas de clé KMS asymétrique pour tenter de chiffrer les données de vos clusters Amazon DocumentDB. Pour plus d'informations, consultez la section Clés asymétriquesAWS KMS dans le Guide duAWS Key Management Service développeur.

Si Amazon DocumentDB ne peut plus accéder à la clé de chiffrement pour un cluster — par exemple, lorsque l'accès à une clé est révoqué — le cluster chiffrée est placée dans un état de mise hors service. Dans ce cas, vous pouvez uniquement restaurer le cluster à partir d'une sauvegarde. Pour Amazon DocumentDB, les sauvegardes sont toujours activées pendant 1 jour.

En outre, si vous désactivez la clé d'un cluster Amazon DocumentDB chiffré, vous finirez par perdre l'accès en lecture et en écriture à ce cluster. Lorsqu'Amazon DocumentDB rencontre un cluster chiffré par une clé à laquelle il n'a pas accès, le cluster passe à l'état de terminal. Dans cet état, le cluster n'est plus disponible et l'état actuel de la base de données ne peut pas être récupéré. Pour restaurer le cluster, vous devez réactiver l'accès à la clé de chiffrement pour Amazon DocumentDB, puis restaurer le cluster à partir d'une sauvegarde.

> **Important**
>
> Vous ne pouvez pas modifier la clé KMS pour un cluster chiffrées après l'avoir déjà créé. Assurez-vous donc de déterminer vos besoins en termes de clés de chiffrement avant de créer votre cluster chiffré.

Using the AWS Management Console

Vous spécifiez l'option chiffrement au repos lorsque vous créez un cluster. Le chiffrement au repos est activé par défaut lorsque vous créez un cluster à l'aide de la AWS Management Console. Il ne peut pas être modifié après la création du cluster.

Pour spécifier l'option de chiffrement au repos, lors de la création de votre cluster

1. Créez un cluster Amazon DocumentDB comme décrit dans la section Mise en route. Toutefois, à l'étape 6, ne choisissez pas Create cluster (Créer un cluster).
2. Sous la section Authentication (Authentification), choisissez Show advanced settings (Afficher les paramètres avancés).
3. Faites défiler jusqu'à lancryption-at-rest section E.
4. Choisissez l'option souhaitée pour le chiffrement au repos. Quelle que soit l'option choisie, vous ne pouvez pas la modifier après la création du cluster.
   - Pour chiffrer les données au repos dans ce cluster, choisissez Enable encryption (Activer le chiffrement).
   - Si vous ne souhaitez pas chiffrer les données au repos dans ce cluster, choisissez Disable encryption (Désactiver le chiffrement).
5. Choisissez la clé principale que vous souhaitez. Amazon DocumentDB utilise leAWS Key Management Service (AWS KMS) pour récupérer et gérer les clés de chiffrement et pour définir les politiques qui contrôlent la manière dont ces clés peuvent être utilisées. Si vous ne spécifiez aucun identifiant deAWS KMS clé, Amazon DocumentDB utilise la clé KMS du serviceAWS géré par défaut. Pour plus d'informations, consultez Concepts AWS Key Management Service.

   > **Note**
   >
   > Après avoir créé un cluster chiffrées, vous ne pouvez pas modifier la clé KMS pour ce cluster. Assurez-vous donc de déterminer vos besoins en termes de clés de chiffrement avant de créer votre cluster chiffré.

6. Complétez les autres sections selon vos besoins et créez votre cluster.

Using the AWS CLI

Pour chiffrer un cluster Amazon DocumentDB à l'aide duAWS CLI, vous devez spécifier l'`--storage-encrypted`option lors de la création du cluster. Les clusters Amazon DocumentDB créés à l'aide du chiffrement du stockageAWS CLI ne sont pas activés par défaut.

L'exemple suivant crée un cluster Amazon DocumentDB avec le chiffrement du stockage activé.

Example

Pour Linux, macOS ou Unix :

```
aws docdb create-db-cluster \
      --db-cluster-identifier sample-cluster \
      --port 27017 \
      --engine docdb \
      --master-username yourMasterUsername \
      --master-user-password yourMasterPassword \
      --storage-encrypted
```

Pour Windows :

```
aws docdb create-db-cluster ^
      --db-cluster-identifier sample-cluster ^
      --port 27017 ^
      --engine docdb ^
      --master-username yourMasterUsername ^
      --master-user-password yourMasterPassword ^
      --storage-encrypted
```

Lorsque vous créez un cluster Amazon DocumentDB chiffré, vous pouvez spécifier un identifiantAWS KMS clé, comme dans l'exemple suivant.

Example

Pour Linux, macOS ou Unix :

```
aws docdb create-db-cluster \
      --db-cluster-identifier sample-cluster \
      --port 27017 \
      --engine docdb \
      --master-username yourMasterUsername \
      --master-user-password yourMasterPassword \
      --storage-encrypted \
      --kms-key-id key-arn-or-alias
```

Pour Windows :

```
aws docdb create-db-cluster ^
```

```
    --db-cluster-identifier sample-cluster ^
    --port 27017 ^
    --engine docdb ^
    --master-username yourMasterUsername ^
    --master-user-password yourMasterPassword ^
    --storage-encrypted ^
    --kms-key-id key-arn-or-alias
```

> **Note**
>
> Après avoir créé un cluster chiffrées, vous ne pouvez pas modifier la clé KMS pour ce cluster. Assurez-vous donc de déterminer vos besoins en termes de clés de chiffrement avant de créer votre cluster chiffré.

Limitations relatives aux clusters chiffrés Amazon DocumentDB

Les limitations suivantes existent pour les limitations suivantes existent pour les limitations suivantes existent pour les clusters chiffrés Amazon DocumentDB

- Vous pouvez activer ou désactiver le chiffrement au repos pour un cluster Amazon DocumentDB uniquement au moment de sa création, et non après la création du cluster. Toutefois, vous pouvez créer une copie chiffrée d'un cluster non chiffré en créant un instantané du cluster non chiffré, puis en restaurant l'instantané non chiffré en tant que nouveau cluster tout en spécifiant l'option de chiffrement au repos.

  Pour plus d'informations, consultez les rubriques suivantes :

  - Création d'un instantané manuel d'un cluster
  - Restauration d'un cluster à partir d'un instantané
  - Copier des instantanés du cluster Amazon DocumentDB
- Les clusters Amazon DocumentDB sur lesquels le chiffrement du stockage est activé ne peuvent pas être modifiés pour désactiver le chiffrement.
- Toutes les instances, les sauvegardes automatisées, les instantanés et les index d'un cluster Amazon DocumentDB sont chiffrés avec la même clé KMS.

Chiffrement des données en transit

Vous pouvez utiliser le protocole TLS (Transport Layer Security) pour chiffrer la connexion entre votre application et un cluster Amazon DocumentDB. Par défaut, le chiffrement en transit est activé pour les clusters Amazon DocumentDB nouvellement créés. Il peut éventuellement être désactivé lors de la création du cluster, ou ultérieurement. Lorsque le chiffrement en transit est activé, des connexions sécurisées à l'aide de TLS sont nécessaires pour se connecter au cluster. Pour plus d'informations sur la connexion à Amazon DocumentDB à l'aide de TLS, veuillez consulter Connexion par programmation à Amazon DocumentDB.

Gestion des paramètres TLS du cluster Amazon DocumentDB

Le chiffrement en transit pour un cluster Amazon DocumentDB est géré via le paramètre TLS dans un groupe de paramètres de cluster. Vous pouvez gérer les paramètres TLS de votre cluster Amazon DocumentDB à l'aide du AWS Management Console ou du AWS Command Line Interface ().AWS CLI Consultez les sections suivantes pour savoir comment vérifier et modifier vos paramètres TLS actuels.

Using the AWS Management Console

Suivez ces étapes pour effectuer des tâches de gestion du chiffrement TLS à l'aide de la console, telles que l'identification des groupes de paramètres, la vérification de la valeur TLS et les modifications nécessaires.

> **Note**
>
> À moins de le spécifier différemment lors de la création d'un cluster, votre cluster est créé avec le groupe de paramètres de cluster par défaut. Les paramètres du groupe de paramètres de cluster `default` ne peuvent pas être modifiés (par exemple, `tls` activé/désactivé). Par conséquent, si votre cluster utilise un groupe de paramètres de cluster `default`, vous devez modifier le cluster pour utiliser un groupe de paramètres de cluster autre que par défaut. Tout d'abord, vous allez peut-être devoir créer un groupe de paramètres de cluster personnalisé. Pour plus d'informations, consultez Création de groupes de paramètres de cluster Amazon DocumentDB.

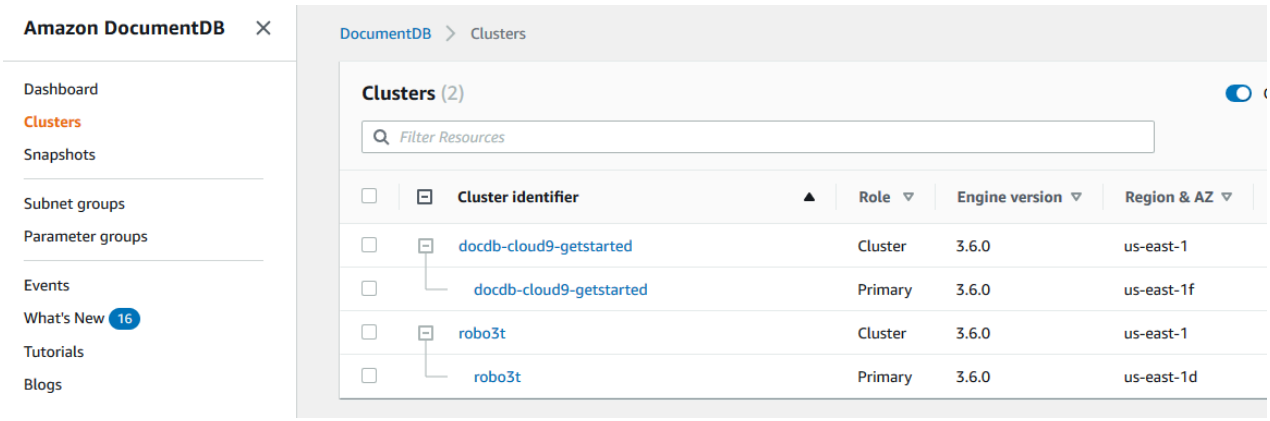
1. Déterminez le groupe de paramètres de cluster utilisé par votre cluster.

a. Ouvrez la console Amazon DocumentDB à l'adresse https://console.aws.amazon.com/docdb.

b. Dans le panneau de navigation, choisissez Clusters.

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

c. Notez que dans la zone de navigation Clusters, la colonne Cluster Identifier indique à la fois les clusters et les instances. Les instances sont répertoriées sous les clusters. Voir la capture d'écran ci-dessous pour référence.



d. Choisissez le cluster qui vous intéresse.

e. Choisissez l'onglet Configuration, faites défiler la page jusqu'en bas de la section Détails du cluster et localisez le groupe de paramètres du cluster. Notez le nom du groupe de paramètres de cluster.

Si le nom du groupe de paramètres du cluster est `default`, par exemple `default.docdb3.6`, vous devez créer un groupe de paramètres de cluster personnalisé et le désigner groupe de paramètres du cluster avant de continuer. Pour plus d'informations, consultez les ressources suivantes :

1. Création de groupes de paramètres de cluster Amazon DocumentDB— Si vous ne disposez pas d'un groupe de paramètres de cluster personnalisé que vous pouvez utiliser, créez-en un.

2. Modification d'un cluster Amazon DocumentDB— Modifiez votre cluster pour utiliser le groupe de paramètres de cluster personnalisé.

2. Déterminez la valeur actuelle du paramètre de cluster **`tls`**.

   a. Ouvrez la console Amazon DocumentDB à l'adresse https://console.aws.amazon.com/docdb.

   b. Dans le panneau de navigation, choisissez Groupes de paramètres.

   c. Dans la liste des groupes de paramètres de cluster, choisissez le nom du groupe qui vous intéresse.

   d. Recherchez la section Cluster parameters (Paramètres de cluster). Dans la liste des paramètres de cluster, recherchez la ligne de paramètre de cluster `tls`. À ce stade, les quatre colonnes suivantes sont importantes :

      - Nom du paramètre du cluster : nom des paramètres du cluster. Pour gérer TLS, intéressez-vous au paramètre de cluster `tls`.
      - Valeurs — La valeur actuelle de chaque paramètre de cluster.
      - Valeurs autorisées : liste de valeurs pouvant être appliquées à un paramètre de cluster.
      - Type d'application : statique ou dynamique. Les modifications apportées aux paramètres de cluster statiques ne peuvent être appliquées que lorsque les instances sont redémarrées. Les modifications apportées aux paramètres de cluster dynamiques peuvent être appliquées immédiatement ou lorsque les instances sont redémarrées.

3. Modifiez la valeur du paramètre de cluster **`tls`**.

   Si la valeur de `tls` n'est pas la valeur requise, modifiez-la pour ce groupe de paramètres de cluster. Pour modifier la valeur du paramètre de cluster `tls`, continuez à partir de la section précédente en suivant les étapes ci-dessous.

   a. Choisissez le bouton à gauche du nom du paramètre de cluster (`tls`).

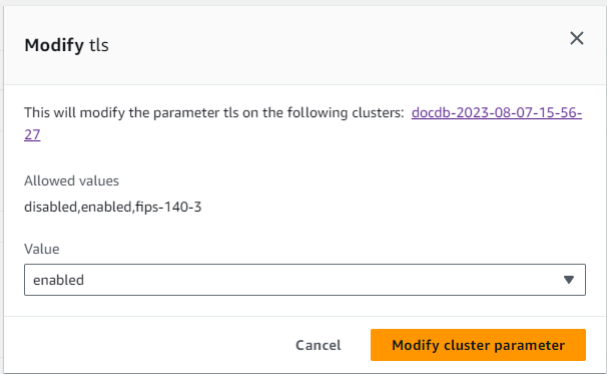
   b. Choisissez Modifier.

   c. Pour modifier la valeur de`tls`, dans la boîte de `tls` dialogue Modifier, choisissez la valeur que vous souhaitez pour le paramètre de cluster dans la liste déroulante.

      Les valeurs valides sont :

      - désactivé — Désactive le protocole TLS

- activé — Active le protocole TLS (versions 1.0, 1.1, 1.2 et 1.3)
- fips-140-3 — Active le protocole TLS avec FIPS. Le cluster accepte uniquement les connexions sécurisées conformément aux exigences de la publication 140-3 des Federal Information Processing Standards (FIPS). Ceci n'est pris en charge qu'à partir des clusters Amazon DocumentDB 5.0 (moteur version 3.0.3727) dans les régions suivantes : ca-central-1, us-west-2, us-east-1, us-east-2, -1, -1. us-gov-east us-gov-west



d. Choisissez Modifier le paramètre de cluster. La modification est appliquée à chaque instance de cluster lors de son redémarrage.

4. Redémarrez l'instance Amazon DocumentDB.

   Redémarrez chaque instance du cluster de sorte que la modification s'applique à toutes les instances du cluster.

   a. Ouvrez la console Amazon DocumentDB à l'adresse https://console.aws.amazon.com/docdb.

   b. Dans le panneau de navigation, sélectionnez Instances.

   c. Pour spécifier une instance à redémarrer, recherchez celle-ci dans la liste des instances et choisissez le bouton à gauche de son nom.

   d. Choisissez Actions, puis Reboot (Redémarrer). Confirmez que vous souhaitez redémarrer en choisissant Reboot (Redémarrer).

Using the AWS CLI

Suivez ces étapes pour effectuer des tâches de gestion du chiffrement TLS à l'aide du, AWS CLI telles que l'identification des groupes de paramètres, la vérification de la valeur TLS et les modifications nécessaires.

> **Note**
>
> À moins de le spécifier différemment lors de la création d'un cluster, le cluster est créé avec le groupe de paramètres de cluster par défaut. Les paramètres du groupe de paramètres de cluster `default` ne peuvent pas être modifiés (par exemple, `tls` activé/désactivé). Par conséquent, si votre cluster utilise un groupe de paramètres de cluster `default`, vous devez modifier le cluster pour utiliser un groupe de paramètres de cluster autre que par défaut. Tout d'abord, vous allez peut-être devoir créer un groupe de paramètres de cluster personnalisé. Pour plus d'informations, consultez Création de groupes de paramètres de cluster Amazon DocumentDB.

1. Déterminez le groupe de paramètres de cluster utilisé par votre cluster.

   Utilisez la commande `describe-db-clusters` avec les paramètres suivants :

   - **`--db-cluster-identifier`** — Obligatoire. Nom du cluster qui vous intéresse.
   - **`--query`**— Facultatif Requête qui limite la sortie aux seuls champs d'intérêt (dans ce cas, le nom du cluster et le nom du groupe de paramètres de cluster).

   ```
   aws docdb describe-db-clusters \
          --db-cluster-identifier docdb-2019-05-07-13-57-08 \
          --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
   ```

   La sortie de cette opération ressemble à ceci (format JSON).

   ```
   [
          [
              "docdb-2019-05-07-13-57-08",
              "custom3-6-param-grp"
          ]
   ]
   ```

Si le nom du groupe de paramètres du cluster est `default`, par exemple `default.docdb3.6`, vous devez disposer d'un groupe de paramètres de cluster personnalisé et le désigner groupe de paramètres du cluster avant de continuer. Pour plus d'informations, consultez les rubriques suivantes :

1. Création de groupes de paramètres de cluster Amazon DocumentDB— Si vous ne disposez pas d'un groupe de paramètres de cluster personnalisé que vous pouvez utiliser, créez-en un.
2. Modification d'un cluster Amazon DocumentDB— Modifiez votre cluster pour utiliser le groupe de paramètres de cluster personnalisé.

2. Déterminez la valeur actuelle du paramètre de cluster **`tls`**.

   Pour obtenir plus d'informations sur ce groupe de paramètres de cluster, utilisez l'opération `describe-db-cluster-parameters` avec les paramètres suivants :

   - **`--db-cluster-parameter-group-name`** — Obligatoire. Utilisez le nom du groupe de paramètres de cluster à partir de la sortie de la commande précédente.
   - **`--query`**— Facultatif Une requête qui limite la sortie aux champs d'intérêt, dans ce cas, le `ParameterName`, `ParameterValue`, `AllowedValues`, et `ApplyType`.

```
aws docdb describe-db-cluster-parameters \
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --query 'Parameters[*].
[ParameterName,ParameterValue,AllowedValues,ApplyType]'
```

   La sortie de cette opération ressemble à ceci (format JSON).

```
[
    [
        "audit_logs",
        "disabled",
        "enabled,disabled",
        "dynamic"
    ],
    [
        "tls",
```

```
        "disabled",
        "disabled,enabled,fips-140-3",
        "static"
    ],
    [
        "ttl_monitor",
        "enabled",
        "disabled,enabled",
        "dynamic"
    ]
]
```

3. Modifiez la valeur du paramètre de cluster **`tls`**.

   Si la valeur de `tls` n'est pas la valeur requise, modifiez-la pour ce groupe de paramètres de cluster. Pour modifier la valeur du paramètre de cluster `tls`, utilisez l'opération `modify-db-cluster-parameter-group` avec les paramètres suivants.

   - **`--db-cluster-parameter-group-name`** — Obligatoire. Nom du groupe de paramètres de cluster à modifier. Il ne peut pas s'agir d'un groupe de paramètres de cluster `default.*`.
   - **`--parameters`** — Obligatoire. Liste des paramètres du groupe de paramètres de cluster à modifier.
     - **`ParameterName`** — Obligatoire. Nom du paramètre de cluster à modifier.
     - **`ParameterValue`** — Obligatoire. Nouvelle valeur pour ce paramètre de cluster. Il doit s'agir de l'une des valeurs `AllowedValues` des paramètres de cluster.
       - **`enabled`**— Le cluster accepte uniquement les connexions sécurisées utilisant les versions TLS 1.0, 1.1, 1.2 ou 1.3.
       - **`disabled`**— Le cluster n'accepte pas les connexions sécurisées utilisant le protocole TLS.
       - **`fips-140-3`**— Le cluster accepte uniquement les connexions sécurisées conformément aux exigences de la publication 140-3 des Federal Information Processing Standards (FIPS). Ceci n'est pris en charge qu'à partir des clusters Amazon DocumentDB 5.0 (moteur version 3.0.3727) dans les régions suivantes : ca-central-1, us-west-2, us-east-1, us-east-2, -1, -1. us-gov-east us-gov-west
     - **`ApplyMethod`**— Quand cette modification doit être appliquée. Pour les paramètres de cluster statiques, tels que `tle`, cette valeur doit être `pending-reboot`.

- **`pending-reboot`**— La modification n'est appliquée à une instance qu'après son redémarrage. Vous devez redémarrer chaque instance de cluster individuellement pour que cette modification soit appliquée sur l'ensemble des instances du cluster.

Le code suivant désactive `tls` et applique la modification à chaque instance de base de données lorsqu'elle est redémarrée.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --parameters "ParameterName=tls,ParameterValue=disabled,ApplyMethod=pending-
reboot"
```

Le code suivant permet `tls` (versions 1.0, 1.1. 1.2 et 1.3) d'appliquer la modification à chaque instance de base de données lors de son redémarrage.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --parameters "ParameterName=tls,ParameterValue=enabled,ApplyMethod=pending-
reboot"
```

Le code suivant active TLS with`fips-140-3`, en appliquant la modification à chaque instance de base de données lors de son redémarrage.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name custom5-0-param-grp \
    --parameters
 "ParameterName=tls,ParameterValue=fips-140-3,ApplyMethod=pending-reboot"
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "DBClusterParameterGroupName": "custom3-6-param-grp"
}
```

4. Redémarrez votre instance Amazon DocumentDB.

   Redémarrez chaque instance du cluster de sorte que la modification s'applique à toutes les instances du cluster. Pour redémarrer une instance Amazon DocumentDB, utilisez l'`reboot-db-instance`opération avec le paramètre suivant :

- **`--db-instance-identifier`** — Obligatoire. Identifiant de l'instance à redémarrer.

Le code suivant redémarre l'instance `sample-db-instance`.

Example

Pour Linux, macOS ou Unix :

```
aws docdb reboot-db-instance \
       --db-instance-identifier sample-db-instance
```

Pour Windows :

```
aws docdb reboot-db-instance ^
       --db-instance-identifier sample-db-instance
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "DBInstance": {
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "PreferredMaintenanceWindow": "fri:09:32-fri:10:02",
        "PendingModifiedValues": {},
        "DBInstanceStatus": "rebooting",
        "DBSubnetGroup": {
            "Subnets": [
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetIdentifier": "subnet-4e26d263"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1c"
                    },
                    "SubnetIdentifier": "subnet-afc329f4"
```

```
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1e"
                    },
                    "SubnetIdentifier": "subnet-b3806e8f"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1d"
                    },
                    "SubnetIdentifier": "subnet-53ab3636"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1b"
                    },
                    "SubnetIdentifier": "subnet-991cb8d0"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1f"
                    },
                    "SubnetIdentifier": "subnet-29ab1025"
                }
            ],
            "SubnetGroupStatus": "Complete",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-91280df6",
            "DBSubnetGroupName": "default"
        },
        "PromotionTier": 2,
        "DBInstanceClass": "db.r5.4xlarge",
        "InstanceCreateTime": "2018-11-05T23:10:49.905Z",
        "PreferredBackupWindow": "00:00-00:30",
        "KmsKeyId": "arn:aws:kms:us-east-1:012345678901:key/0961325d-a50b-44d4-
b6a0-a177d5ff730b",
        "StorageEncrypted": true,
        "VpcSecurityGroups": [
            {
```

```
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "EngineVersion": "3.6.0",
        "DbiResourceId": "db-SAMPLERESOURCEID",
        "DBInstanceIdentifier": "sample-cluster-instance-00",
        "Engine": "docdb",
        "AvailabilityZone": "us-east-1a",
        "DBInstanceArn": "arn:aws:rds:us-east-1:012345678901:db:sample-cluster-
instance-00",
        "BackupRetentionPeriod": 1,
        "Endpoint": {
            "Address": "sample-cluster-instance-00.corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
            "Port": 27017,
            "HostedZoneId": "Z2R2ITUGPM61AM"
        },
        "DBClusterIdentifier": "sample-cluster"
    }
}
```

Le redémarrage de votre instance prend quelques minutes. Vous pouvez uniquement utiliser l'instance lorsqu'elle présente le statut disponible. Vous pouvez surveiller l'état de l'instance en utilisant la console ou la AWS CLI. Pour plus d'informations, voir Surveillance de l'état d'une instance Amazon DocumentDB.

Gestion des clés

Amazon DocumentDB utilise AWS Key Management Service (AWS KMS) pour récupérer et gérer les clés de chiffrement. AWS KMScombine du matériel et des logiciels sécurisés et hautement disponibles pour fournir un système de gestion des clés adapté au cloud. En utilisant AWS KMS, vous pouvez créer des clés de chiffrement et définir les politiques qui contrôlent la manière dont ces clés peuvent être utilisées. AWS KMS prend en charge AWS CloudTrail, afin de vous permettre de vérifier que l'utilisation des clés est appropriée.

Vos AWS KMS clés peuvent être utilisées en combinaison avec Amazon DocumentDB et les AWS services pris en charge tels qu'Amazon Simple Storage Service (Amazon S3), Amazon Relational Database Service (Amazon RDS), Amazon Elastic Block Store (Amazon EBS) et Amazon Redshift. Pour obtenir la liste des services compatiblesAWS KMS, consultez la section Comment les AWS

services sont utilisés AWS KMS dans le Guide du AWS Key Management Service développeur. Pour plus d'informations sur AWS KMS, consultez Qu'est-ce que AWS Key Management Service ?

Identity and Access Management pour Amazon DocumentDB

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources Amazon DocumentDB. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez dans Amazon DocumentDB.

Utilisateur du service : si vous utilisez le service Amazon DocumentDB pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de plus en plus de fonctionnalités d'Amazon DocumentDB pour effectuer votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations

appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité dans Amazon DocumentDB, consultez. Résolution des problèmes d'identité et d'accès à Amazon DocumentDB

Administrateur de service — Si vous êtes responsable des ressources Amazon DocumentDB au sein de votre entreprise, vous avez probablement un accès complet à Amazon DocumentDB. C'est à vous de déterminer les fonctionnalités et les ressources Amazon DocumentDB auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser IAM avec Amazon DocumentDB, consultez. Comment Amazon DocumentDB fonctionne avec IAM

Administrateur IAM : si vous êtes administrateur IAM, vous souhaiterez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès à Amazon DocumentDB. Pour consulter des exemples de politiques basées sur l'identité Amazon DocumentDB que vous pouvez utiliser dans IAM, consultez. Exemples de politiques basées sur l'identité pour Amazon DocumentDB

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 (connecté à AWS) en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section Comment vous connecter à votre compte Compte AWS dans le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-

même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vous-même les demandes, consultez la section Signature des demandes AWS d'API dans le guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir plus, consultez Authentification multifactorielle dans le Guide de l'utilisateur AWS IAM Identity Center et Utilisation de l'authentification multifactorielle (MFA) dans l'interface AWS dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur root, consultez Tâches nécessitant des informations d'identification d'utilisateur root dans le Guide de l'utilisateur IAM.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris ceux qui ont besoin d'un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles assument des rôles, qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour

obtenir des informations sur IAM Identity Center, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification dans le Guide de l'utilisateur IAM.

Un groupe IAM est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé IAMAdmins et accorder à ce groupe les autorisations d'administrer des ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, consultez Quand créer un utilisateur IAM (au lieu d'un rôle) dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité au sein de vous Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un rôle IAM dans le en AWS Management Console changeant de rôle. Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez Utilisation de rôles IAM dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré – Pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour

obtenir des informations sur les rôles pour la fédération, consultez Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez la rubrique Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .

- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
  - Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez Sessions de transmission d'accès.
  - Rôle de service : il s'agit d'un rôle IAM attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.
  - Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés

à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une instance EC2 et qui envoient des demandes d'API. AWS CLI AWS Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un AWS rôle à une instance EC2 et le mettre à la disposition de toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2 dans le Guide de l'utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, consultez Quand créer un rôle IAM (au lieu d'un utilisateur) dans le Guide de l'utilisateur IAM.

Gestion des 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 est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

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.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui

autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Création de politiques IAM dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique 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. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. 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. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

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 (ACL)

Les listes de contrôle d'accès (ACL) vérifie quels principals (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux

politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et Amazon VPC sont des exemples de services qui prennent en charge les ACL. AWS WAF Pour en savoir plus sur les listes de contrôle d'accès, consultez Vue d'ensemble des listes de contrôle d'accès (ACL) dans le Guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- Limite d'autorisations : une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCP) — Les SCP sont des politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans. AWS Organizations AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer les politiques de contrôle des services (SCP) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les organisations et les SCP, consultez Fonctionnement des SCP dans le Guide de l'utilisateur AWS Organizations .
- Politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations, consultez politiques de séance 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.

Comment Amazon DocumentDB fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à Amazon DocumentDB, découvrez quelles fonctionnalités IAM peuvent être utilisées avec Amazon DocumentDB.

Fonctionnalités IAM que vous pouvez utiliser avec Amazon DocumentDB

Fonction IAM	Clusters basés sur des instances	Clusters élastiques
Politiques basées sur l'identité	Oui	Oui
Politiques basées sur les ressources	Non	Non
Actions de politique	Oui	Oui
Ressources de politique	Oui	Oui
Clés de condition de politique (spécifiques au service)	Oui	Oui
ACL	Non	Non
ABAC (identifications dans les politiques)	Partielle	Oui
Informations d'identification temporaires	Oui	Oui
Autorisations de principal	Oui	Oui
Fonctions de service	Oui	Oui

Fonction IAM	Clusters basés sur des instances	Clusters élastiques
Rôles liés à un service	Non	Oui

Pour obtenir une vue d'ensemble de la façon dont Amazon DocumentDB et les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez les AWS services compatibles avec IAM dans le guide de l'utilisateur IAM.

Politiques basées sur l'identité pour Amazon DocumentDB

Prend en charge les politiques basées sur l’identité	Oui

Les politiques basées sur l’identité sont des documents de politique d’autorisations JSON que vous pouvez attacher à une identité telle qu’un utilisateur, un groupe d’utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d’actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l’identité, consultez Création de politiques IAM dans le Guide de l’utilisateur IAM.

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. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité car celle-ci s’applique à l’utilisateur ou au rôle auquel elle est attachée. Pour découvrir 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.

Exemples de politiques basées sur l'identité pour Amazon DocumentDB

Pour consulter des exemples de politiques basées sur l'identité Amazon DocumentDB, consultez. Exemples de politiques basées sur l'identité pour Amazon DocumentDB

Politiques basées sur les ressources au sein d'Amazon DocumentDB

Prend en charge les politiques basées sur les ressources	Non

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. 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. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. L'ajout d'un principal entre comptes à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un administrateur IAM du compte sécurisé doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, consultez Différence entre les rôles IAM et les politiques basées sur une ressource dans le Guide de l'utilisateur IAM.

Actions politiques pour Amazon DocumentDB

Prend en charge les actions de politique	Oui

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. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

> **Note**
>
> Pour certaines fonctionnalités de gestion, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon Relational Database Service (Amazon RDS).
> Pour consulter la liste des actions RDS, consultez la section Actions définies par Amazon Relational Database Service dans le Service Authorization Reference.
> Pour consulter les actions politiques relatives aux clusters élastiques Amazon DocumentDB, consultez la section Actions définies par les clusters élastiques Amazon DocumentDB dans la référence d'autorisation de service.

Les actions politiques dans Amazon DocumentDB utilisent le préfixe suivant avant l'action :

```
aws
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
      "aws:action1",
      "aws:action2"
         ]
```

Pour consulter des exemples de politiques basées sur l'identité Amazon DocumentDB, consultez. Exemples de politiques basées sur l'identité pour Amazon DocumentDB

Ressources relatives aux politiques pour Amazon DocumentDB

Prend en charge les ressources de politique	Oui

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. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

> **Note**
>
> Pour certaines fonctionnalités de gestion, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon Relational Database Service (Amazon RDS).
> Pour consulter la liste des types de ressources RDS et de leurs ARN, consultez la section Ressources définies par Amazon Relational Database Service dans le Service Authorization Reference. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par Amazon Relational Database Service.
> Pour consulter les types de ressources pour les clusters élastiques Amazon DocumentDB, consultez la section Types de ressources définis par les clusters élastiques Amazon DocumentDB dans la référence d'autorisation de service.

Pour consulter des exemples de politiques basées sur l'identité Amazon DocumentDB, consultez. Exemples de politiques basées sur l'identité pour Amazon DocumentDB

Clés de conditions de politique pour Amazon DocumentDB

Prend en charge les clés de condition de politique spécifiques au service	Oui

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` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. 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.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez Éléments d'une politique IAM : variables et identifications dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. 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.

> **Note**
>
> Pour certaines fonctionnalités de gestion, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon Relational Database Service (Amazon RDS). Pour consulter la liste des clés de condition RDS, consultez la section Clés de condition pour Amazon Relational Database Service dans le Service Authorization Reference. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez Actions définies par Amazon Relational Database Service.
> Pour consulter les clés de condition pour les clusters élastiques Amazon DocumentDB, consultez la section Clés de condition pour les clusters élastiques Amazon DocumentDB dans la référence d'autorisation de service.

Pour consulter des exemples de politiques basées sur l'identité Amazon DocumentDB, consultez. Exemples de politiques basées sur l'identité pour Amazon DocumentDB

ACL dans Amazon DocumentDB

Prend en charge les listes ACL	Non

Les listes de contrôle d'accès (ACL) vérifient quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

ABAC avec Amazon DocumentDB

> **Note**
>
> ABAC n'est que partiellement pris en charge pour les clusters basés sur des instances, mais il est pris en charge pour les clusters élastiques.

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L'étiquetage des entités et des ressources est la première étape d'ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l'identification du principal correspond à celle de la ressource à laquelle il tente d'accéder.

L'ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

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 `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur l'ABAC, consultez Qu'est-ce que le contrôle d'accès basé sur les attributs (ABAC) ? dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les

étapes de configuration de l'ABAC, consultez Utilisation du contrôle d'accès par attributs (ABAC) dans le Guide de l'utilisateur IAM.

Utilisation d'informations d'identification temporaires avec Amazon DocumentDB

Prend en charge les informations d'identification temporaires	Oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation d'IAM dans le guide de l'utilisateur d'IAM.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez Changement de rôle (console) dans le Guide de l'utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez Informations d'identification de sécurité temporaires dans IAM.

Autorisations principales interservices pour Amazon DocumentDB

Prend en charge les sessions d'accès direct (FAS)	Oui

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux

services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d'accès.

Rôles de service pour Amazon DocumentDB

Prend en charge les fonctions du service	Oui

Une fonction de service est un rôle IAM qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

> **Warning**
>
> La modification des autorisations associées à un rôle de service peut perturber les fonctionnalités d'Amazon DocumentDB. Modifiez les rôles de service uniquement lorsque Amazon DocumentDB fournit des instructions à cet effet.

Rôles liés à un service pour Amazon DocumentDB

> **Note**
>
> Les rôles liés à un service ne sont pas pris en charge pour les clusters basés sur des instances, mais sont pris en charge pour les clusters élastiques.

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur la création ou la gestion des rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM. Recherchez un service dans le tableau qui inclut un `Yes` dans la

colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

Exemples de politiques basées sur l'identité pour Amazon DocumentDB

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier les ressources Amazon DocumentDB. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l'API AWS Management Console, AWS Command Line Interface (AWS CLI) ou de AWS l'API. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez Création de politiques dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par Amazon DocumentDB, y compris le format des ARN pour chacun des types de ressources, consultez la section Actions, ressources et clés de condition pour Amazon Relational Database Service dans la référence d'autorisation du service.

Rubriques

- Bonnes pratiques en matière de politiques
- Utilisation de la console Amazon DocumentDB
- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer des ressources Amazon DocumentDB 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.

- Accorder 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 de IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utiliser 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 AWS CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.
- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : IAM Access Analyzer 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 politique 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 le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Configuration de l'accès aux API protégé par 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 DocumentDB

Pour accéder à la console Amazon DocumentDB (compatible avec MongoDB), 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 DocumentDB présentes dans

votre. Compte AWS Si vous créez une stratégie basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette stratégie.

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 qu'ils tentent d'effectuer.

Pour garantir que les utilisateurs et les rôles peuvent toujours utiliser la console Amazon DocumentDB, associez également Amazon *ConsoleAccess* DocumentDB *ReadOnly* AWS ou la politique gérée aux entités. Pour plus d'informations, consultez Ajout d'autorisations à un utilisateur dans le Guide de l'utilisateur IAM.

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": "*"
        }
    ]
}
```

Résolution des problèmes d'identité et d'accès à Amazon DocumentDB

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec Amazon DocumentDB et IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans Amazon DocumentDB
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures à moi Compte AWS à accéder à mes ressources Amazon DocumentDB

Je ne suis pas autorisé à effectuer une action dans Amazon DocumentDB

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit quand l'utilisateur IAM `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource *`my-example-widget`* fictive, mais ne dispose pas des autorisations `aws:`*`GetWidget`* fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 aws:GetWidget on resource: my-example-widget
```

Dans ce cas, la politique qui s'applique à l'utilisateur `mateojackson` doit être mise à jour pour autoriser l'accès à la ressource *`my-example-widget`* à l'aide de l'action `aws:`*`GetWidget`*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à Amazon DocumentDB.

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, un utilisateur doit 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 DocumentDB. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer 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 moi Compte AWS à accéder à mes ressources Amazon DocumentDB

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 (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si Amazon DocumentDB prend en charge ces fonctionnalités, consultez. Comment Amazon DocumentDB fonctionne 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 découvrir quelle est la différence entre l'utilisation des rôles et l'utilisation des politiques basées sur les ressources pour l'accès entre comptes, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.

Gestion des autorisations d'accès à vos ressources Amazon DocumentDB

Chaque AWS ressource appartient à un Compte AWS, et les autorisations de création ou d'accès aux ressources sont régies par des politiques d'autorisation. Un administrateur de compte peut associer des politiques d'autorisations aux identités IAM (c'est-à-dire aux utilisateurs, aux groupes et aux rôles), et certains services (tels que AWS Lambda) prennent également en charge l'attachement de politiques d'autorisations aux ressources.

> **Note**
>
> Un administrateur de compte (ou utilisateur administrateur) est un utilisateur doté d'autorisations d'administrateur. Pour plus d'informations, consultez Bonnes pratiques IAM dans le Guide de l'utilisateur IAM.

Rubriques

Ressources et opérations Amazon DocumentDB

Dans Amazon DocumentDB, la ressource principale est un cluster. Amazon DocumentDB prend en charge d'autres ressources qui peuvent être utilisées avec la ressource principale, telles que les instances, les groupes de paramètres et les abonnements aux événements. Ces ressources sont appelées sous-ressources.

Ces ressources et sous-ressources ont un ARN (Amazon Resource Name) unique qui leur est associé, comme illustré dans le tableau suivant.

Type de ressource	Format ARN
Cluster	`arn:aws:rds:` *`region`*`:`*`account-id`* `:cluster:` *`db-cluster-name`*
Groupe de paramètres du cluster	`arn:aws:rds:` *`region`*`:`*`account-id`* `:cluster-pg:` *`cluster-parameter-group-name`*
Instantané du cluster	`arn:aws:rds:` *`region`*`:`*`account-id`* `:cluster-snapshot:` *`cluster-snapshot-name`*
Instance	`arn:aws:rds:` *`region`*`:`*`account-id`* `:db:`*`db-instance-name`*
Groupe de sécurité	`arn:aws:rds:` *`region`*`:`*`account-id`* `:secgrp:`*`security-group-name`*
Groupe de sous-réseaux	`arn:aws:rds:` *`region`*`:`*`account-id`* `:subgrp:`*`subnet-group-name`*

Amazon DocumentDB fournit un ensemble d'opérations permettant de travailler avec les ressources Amazon DocumentDB. Pour obtenir la liste des opérations disponibles, consultez Actions.

Présentation de la propriété des ressources

Le propriétaire d'une ressource est celui Compte AWS qui a créé une ressource. En d'autres termes, le propriétaire Compte AWS de la ressource est l'entité principale (le compte root, un utilisateur IAM ou un rôle IAM) qui authentifie la demande qui crée la ressource. Les exemples suivants illustrent comment cela fonctionne :

- Si vous utilisez les informations d'identification de votre compte root Compte AWS pour créer une ressource Amazon DocumentDB, telle qu'une instance, vous Compte AWS êtes le propriétaire de la ressource Amazon DocumentDB.
- Si vous créez un utilisateur IAM dans votre compte Compte AWS et que vous accordez l'autorisation de créer des ressources Amazon DocumentDB à cet utilisateur, celui-ci peut créer des ressources Amazon DocumentDB. Cependant, c'est à vous Compte AWS, à laquelle appartient l'utilisateur, que vous êtes propriétaire des ressources Amazon DocumentDB.
- Si vous créez un rôle IAM Compte AWS avec l'autorisation de créer des ressources Amazon DocumentDB, toute personne capable d'assumer ce rôle peut créer des ressources Amazon DocumentDB. À qui appartient le rôle Compte AWS, vous êtes propriétaire des ressources Amazon DocumentDB.

Gestion de l'accès aux ressources

Une politique d'autorisation décrit qui a accès à quoi. La section suivante explique les options disponibles pour créer des politiques d'autorisations.

> **Note**
>
> Cette section décrit l'utilisation d'IAM dans le contexte d'Amazon DocumentDB. Elle ne fournit pas d'informations détaillées sur le service IAM. Pour une documentation complète sur IAM, veuillez consulter Qu'est-ce qu'IAM ? dans le Guide de l'utilisateur IAM. Pour plus d'informations sur la syntaxe et les descriptions des politiques IAM, consultez la section Référence des AWSIAM politiques dans le guide de l'utilisateur IAM.

Les politiques qui sont associées à une identité IAM sont appelées des politiques basées sur l'identité (politiques IAM). Les politiques qui sont attachées à une ressource sont appelées politiques basées sur la ressource. Amazon DocumentDB prend uniquement en charge les politiques basées sur l'identité (politiques IAM).

Rubriques

- Politiques basées sur une identité (politiques IAM)
- Politiques basées sur une ressource

Politiques basées sur une identité (politiques IAM)

Vous pouvez attacher des politiques à des identités IAM. Par exemple, vous pouvez effectuer les opérations suivantes :

- Associez une politique d'autorisations à un utilisateur ou à un groupe de votre compte : un administrateur de compte peut utiliser une politique d'autorisations associée à un utilisateur particulier pour autoriser cet utilisateur à créer une ressource Amazon DocumentDB, telle qu'une instance.
- Attacher une politique d'autorisations à un rôle (accorder des autorisations entre comptes) : vous pouvez attacher une politique d'autorisation basée sur une identité à un rôle IAM afin d'accorder des autorisations entre comptes. Par exemple, un administrateur peut créer un rôle pour accorder des autorisations entre comptes à un autre Compte AWS ou à un AWS service comme suit :
  1. L'administrateur du Compte A crée un rôle IAM et attache une politique d'autorisation à ce rôle qui accorde des autorisations sur les ressources dans le Compte A.
  2. L'administrateur du Compte A attache une politique d'approbation au rôle identifiant le Compte B comme principal pouvant assumer ce rôle.
  3. L'administrateur du compte B peut ensuite déléguer les autorisations nécessaires pour assumer le rôle à n'importe quel utilisateur du compte B. Cela permet aux utilisateurs du compte B de créer ou d'accéder aux ressources du compte A. Le principal de la politique de confiance peut également être un principal de AWS service si vous souhaitez autoriser un AWS service à assumer ce rôle.

  Pour en savoir plus sur l'utilisation d'IAM pour déléguer des autorisations, consultez Gestion des accès dans le Guide de l'utilisateur IAM.

Voici un exemple de politique qui permet à l'utilisateur possédant l'ID `123456789012` de créer des instances pour votre Compte AWS. La nouvelle instance doit utiliser un groupe d'options et un groupe de paramètres de base de données commençant par `default`, et elle doit utiliser le groupe de sous-réseaux `default`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateDBInstanceOnly",
            "Effect": "Allow",
```

```
            "Action": [
                "rds:CreateDBInstance"
            ],
            "Resource": [
                "arn:aws:rds:*:123456789012:db:test*",
                "arn:aws:rds:*:123456789012:pg:cluster-pg:default*",
                "arn:aws:rds:*:123456789012:subgrp:default"
            ]
        }
    ]
}
```

Pour plus d'informations sur l'utilisation de politiques basées sur l'identité avec Amazon DocumentDB, consultez. Utilisation de politiques basées sur l'identité (politiques IAM) pour Amazon DocumentDB Pour de plus amples informations sur les utilisateurs, les groupes, les rôles et les autorisations, consultez Identités (utilisateurs, groupes et rôles) dans le Guide de l'utilisateur IAM.

Politiques basées sur une ressource

D'autres services, tels qu'Amazon Simple Storage Service (Amazon S3), prennent également en charge les politiques d'autorisation basées sur les ressources. Par exemple, vous pouvez attacher une politique à un compartiment Amazon S3 pour gérer les autorisations d'accès à ce compartiment. Amazon DocumentDB ne prend pas en charge les politiques basées sur les ressources.

Spécification des éléments d'une stratégie : actions, effets, ressources et mandataires

Pour chaque ressource Amazon DocumentDB (voirRessources et opérations Amazon DocumentDB), le service définit un ensemble d'opérations d'API. Pour plus d'informations, consultez Actions. Pour accorder des autorisations pour ces opérations d'API, Amazon DocumentDB définit un ensemble d'actions que vous pouvez spécifier dans une politique. Une opération d’API peut exiger des autorisations pour plusieurs actions.

Voici les éléments de base d’une politique :

- Ressource : dans une politique, vous utilisez un Amazon Resource Name (ARN) pour identifier la ressource à laquelle la politique s’applique.
- Action : vous utilisez des mots clés d’action pour identifier les opérations de ressource que vous voulez accorder ou refuser. Par exemple, l'autorisation `rds:DescribeDBInstances` permet à l'utilisateur d'effectuer l'opération `DescribeDBInstances`.

- Effet – Vous spécifiez l'effet produit lorsque l'utilisateur demande l'action spécifique, qui peut être une autorisation ou un refus. Si vous n'accordez pas explicitement l'accès pour (autoriser) une ressource, l'accès est implicitement refusé. Vous pouvez aussi explicitement refuser l'accès à une ressource, ce que vous pouvez faire afin de vous assurer qu'un utilisateur n'y a pas accès, même si une politique différente accorde l'accès.
- Principal – dans les politiques basées sur une identité (politiques IAM), l'utilisateur auquel la politique est attachée est le principal implicite. Pour les politiques basées sur une ressource, vous spécifiez l'utilisateur, le compte, le service ou une autre entité qui doit recevoir les autorisations (s'applique uniquement aux politiques basées sur une ressource). Amazon DocumentDB ne prend pas en charge les politiques basées sur les ressources.

Pour en savoir plus sur la syntaxe des stratégies IAM et pour obtenir des descriptions, consultez Référence de stratégie IAM AWS dans le Guide de l'utilisateur IAM.

Pour consulter un tableau présentant toutes les actions de l'API Amazon DocumentDB et les ressources auxquelles elles s'appliquent, consultez. Autorisations d'API Amazon DocumentDB : référence des actions, des ressources et des conditions

Spécification de conditions dans une politique

Lorsque vous accordez des autorisations, vous pouvez utiliser le langage des politiques IAM afin de spécifier les conditions définissant à quel moment une politique doit prendre effet. Par exemple, il est possible d'appliquer une politique après seulement une date spécifique. Pour plus d'informations sur la spécification de conditions dans un langage de politique, consultez Condition dans le Guide de l'utilisateur IAM.

Pour exprimer des conditions, vous utilisez des clés de condition prédéfinies. Amazon DocumentDB ne possède aucune clé de contexte spécifique à un service pouvant être utilisée dans une politique IAM. Pour accéder à la liste des clés de contexte de condition disponibles pour tous les services, consultez Clés de condition disponibles dans le Guide de l'utilisateur IAM.

Utilisation de politiques basées sur l'identité (politiques IAM) pour Amazon DocumentDB

> **Important**
>
> Pour certaines fonctionnalités de gestion, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon RDS. Les appels à la console Amazon DocumentDB et à l'API sont enregistrés en tant qu'appels passés à l'API Amazon RDS. AWS CLI
> Nous vous recommandons de consulter d'abord les rubriques d'introduction qui expliquent les concepts de base et les options disponibles pour gérer l'accès à vos ressources Amazon DocumentDB. Pour plus d'informations, consultez Gestion des autorisations d'accès à vos ressources Amazon DocumentDB.

Cette rubrique fournit des exemples de politiques basées sur une identité dans lesquelles un administrateur de compte peut attacher des politiques d'autorisation aux identités IAM (c'est-à-dire aux utilisateurs, groupes et rôles).

Voici un exemple de politique IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateDBInstanceOnly",
            "Effect": "Allow",
            "Action": [
                "rds:CreateDBInstance"
            ],
            "Resource": [
                "arn:aws:rds:*:123456789012:db:test*",
                "arn:aws:rds:*:123456789012:pg:cluster-pg:default*",
                "arn:aws:rds:*:123456789012:subgrp:default"
            ]
        }
    ]
}
```

La politique inclut une instruction unique spécifiant les autorisations suivantes pour l'utilisateur IAM :

- La politique permet à l'utilisateur IAM de créer une instance à l'aide de l'action CreateDBInstance (cela s'applique également à l'opération et create-db-instance AWS CLI au). AWS Management Console
- L'élément `Resource` spécifie que l'utilisateur peut effectuer des actions sur et avec des ressources. Vous indiquez des ressources à l'aide d'un Amazon Resources Name (ARN). Cet ARN inclut le nom du service auquel appartient la ressource (`rds`), le Région AWS (`*`indique n'importe quelle région dans cet exemple), le numéro de compte utilisateur (`123456789012`il s'agit de l'ID utilisateur dans cet exemple) et le type de ressource.

  L'élément `Resource` dans l'exemple spécifie les contraintes de stratégie suivantes sur les ressources de l'utilisateur :

  - L'identifiant d'instance de la nouvelle instance doit commencer par `test` (par exemple, `testCustomerData1`, `test-region2-data`).
  - Le groupe de paramètres du cluster de la nouvelle instance doit commencer par `default`.
  - Le groupe de sous-réseaux de la nouvelle instance doit être le groupe de sous-réseaux `default`.

La politique ne spécifie pas l'élément `Principal`, car dans une politique basée sur une identité, vous ne spécifiez pas le principal qui obtient l'autorisation. Quand vous attachez une politique à un utilisateur, l'utilisateur est le principal implicite. Lorsque vous attachez une politique d'autorisation à un rôle IAM, le principal identifié dans la politique d'approbation de ce rôle obtient les autorisations.

Pour consulter un tableau présentant toutes les opérations de l'API Amazon DocumentDB et les ressources auxquelles elles s'appliquent, consultez. Autorisations d'API Amazon DocumentDB : référence des actions, des ressources et des conditions

Autorisations requises pour utiliser la console Amazon DocumentDB

Pour qu'un utilisateur puisse utiliser la console Amazon DocumentDB, il doit disposer d'un ensemble minimal d'autorisations. Ces autorisations permettent à l'utilisateur de décrire les ressources Amazon DocumentDB qui lui sont associées Compte AWS et de fournir d'autres informations connexes, notamment des informations relatives à la sécurité et au réseau Amazon EC2.

Si vous créez une politique IAM plus restrictive que les autorisations minimales requises, la console ne fonctionnera pas comme prévu pour les utilisateurs dotés de cette politique IAM. Pour garantir que ces utilisateurs peuvent toujours utiliser la console Amazon DocumentDB, associez également

la politique `AmazonDocDBConsoleFullAccess` gérée à l'utilisateur, comme décrit dans. AWS politiques gérées pour Amazon DocumentDB

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API Amazon DocumentDB AWS CLI ou l'API Amazon DocumentDB.

Exemples de politiques gérées par le client

Dans cette section, vous trouverez des exemples de politiques utilisateur qui accordent des autorisations pour diverses actions Amazon DocumentDB. Ces politiques fonctionnent lorsque vous utilisez des actions d'API Amazon DocumentDB, des AWS SDK ou le. AWS CLI Lorsque vous utilisez la console, vous devez accorder des autorisations supplémentaires spécifiques à la console, ce qui est détaillé dans Autorisations requises pour utiliser la console Amazon DocumentDB.

Pour certaines fonctionnalités de gestion, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon Relational Database Service (Amazon RDS) et Amazon Neptune.

> **Note**
>
> Tous les exemples utilisent la région USA Est (Virginie du Nord) (`us-east-1`) et contiennent des identifiants de compte fictifs.

Exemples

- Exemple 1 : autoriser un utilisateur à effectuer n'importe quelle action de description sur n'importe quelle ressource Amazon DocumentDB
- Exemple 2 : Empêcher un utilisateur de supprimer une instance
- Exemple 3 : Empêcher un utilisateur de créer un cluster à moins que le chiffrement du stockage ne soit activé

Exemple 1 : autoriser un utilisateur à effectuer n'importe quelle action de description sur n'importe quelle ressource Amazon DocumentDB

La politique d'autorisation suivante accorde des autorisations à un utilisateur lui permettant d'exécuter toutes les actions commençant par `Describe`. Ces actions affichent des informations sur une ressource Amazon DocumentDB, telle qu'une instance. Le caractère générique (*) dans

l'`Resource`élément indique que les actions sont autorisées pour toutes les ressources Amazon DocumentDB détenues par le compte.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowRDSDescribe",
         "Effect":"Allow",
         "Action":"rds:Describe*",
         "Resource":"*"
      }
   ]
}
```

Exemple 2 : Empêcher un utilisateur de supprimer une instance

La stratégie d'autorisation suivante accorde des autorisations empêchant un utilisateur de supprimer une instance spécifique. Par exemple, il est possible de refuser la capacité à supprimer vos instances de production à un utilisateur quelconque qui n'est pas un administrateur.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"DenyDelete1",
         "Effect":"Deny",
         "Action":"rds:DeleteDBInstance",
         "Resource":"arn:aws:rds:us-east-1:123456789012:db:my-db-instance"
      }
   ]
}
```

Exemple 3 : Empêcher un utilisateur de créer un cluster à moins que le chiffrement du stockage ne soit activé

La politique d'autorisation suivante interdit à un utilisateur de créer un cluster Amazon DocumentDB à moins que le chiffrement du stockage ne soit activé.

```
{
   "Version": "2012-10-17",
   "Statement": [
```

```
      {
         "Sid": "PreventUnencryptedDocumentDB",
         "Effect": "Deny",
         "Action": "RDS:CreateDBCluster",
         "Condition": {
         "Bool": {
         "rds:StorageEncrypted": "false"
      },
         "StringEquals": {
         "rds:DatabaseEngine": "docdb"
         }
      },
      "Resource": "*"
      }
   ]
}
```

AWS politiques gérées pour Amazon DocumentDB

Pour ajouter des autorisations aux utilisateurs, aux groupes et aux rôles, il est plus facile d'utiliser des politiques AWS gérées que de les rédiger vous-même. Il faut du temps et de l'expertise pour créer des politiques gérées par le client IAM qui ne fournissent à votre équipe que les autorisations dont elle a besoin. Pour démarrer rapidement, vous pouvez utiliser nos politiques AWS gérées. Ces politiques couvrent les cas d'utilisation courants et sont disponibles dans votre AWS compte. Pour plus d'informations sur les politiques AWS gérées, consultez les politiques AWS gérées dans le guide de l'utilisateur d'AWS Identity and Access Management.

AWS les services maintiennent et mettent à jour les politiques AWS gérées. Vous ne pouvez pas modifier les autorisations dans les politiques AWS gérées. Les services ajoutent parfois des autorisations supplémentaires à une politique AWS gérée pour prendre en charge de nouvelles fonctionnalités. Ce type de mise à jour affecte toutes les identités (utilisateurs, groupes et rôles) auxquelles la politique est attachée. Les services sont plus susceptibles de mettre à jour une politique AWS gérée lorsqu'une nouvelle fonctionnalité est lancée ou lorsque de nouvelles opérations sont disponibles. Les services ne suppriment pas les autorisations d'une politique AWS gérée. Les mises à jour des politiques n'endommageront donc pas vos autorisations existantes.

En outre, AWS prend en charge les politiques gérées pour les fonctions professionnelles qui couvrent plusieurs services. Par exemple, la politique `ViewOnlyAccess` AWS gérée fournit un accès en lecture seule à de nombreux AWS services et ressources. Lorsqu'un service lance une nouvelle fonctionnalité, il AWS ajoute des autorisations en lecture seule pour les nouvelles opérations et

ressources. Pour obtenir une liste et une description des politiques relatives aux fonctions de travail, voir les politiques AWS gérées pour les fonctions de travail dans le guide de l'utilisateur d'AWS Identity and Access Management.

Les politiques AWS gérées suivantes, que vous pouvez associer aux utilisateurs de votre compte, sont spécifiques à Amazon DocumentDB :

- AmazonDocDB FullAccess— Accorde un accès complet à toutes les ressources Amazon DocumentDB pour le compte root AWS .
- AmazonDocDB ReadOnlyAccess— Accorde un accès en lecture seule à toutes les ressources Amazon DocumentDB pour le compte root. AWS
- AmazonDocDB ConsoleFullAccess— Accorde un accès complet pour gérer les ressources du cluster élastique Amazon DocumentDB et Amazon DocumentDB à l'aide du. AWS Management Console
- AmazonDocDB ElasticReadOnlyAccess— Accorde un accès en lecture seule à toutes les ressources du cluster élastique Amazon DocumentDB pour le compte racine. AWS
- AmazonDocDB ElasticFullAccess— Accorde un accès complet à toutes les ressources du cluster élastique Amazon DocumentDB pour le compte root AWS .

AmazonDocDB FullAccess

Cette politique accorde des autorisations administratives qui permettent un accès complet principal à toutes les actions Amazon DocumentDB. Les autorisations définies dans cette politique sont regroupées comme suit :

- Les autorisations Amazon DocumentDB autorisent toutes les actions Amazon DocumentDB.
- Certaines des autorisations Amazon EC2 définies dans cette politique sont requises pour valider les ressources transmises dans le cadre d'une demande d'API. Cela permet de s'assurer qu'Amazon DocumentDB est capable d'utiliser correctement les ressources avec un cluster. Les autres autorisations Amazon EC2 de cette politique permettent à Amazon DocumentDB de AWS créer les ressources nécessaires pour vous permettre de vous connecter à vos clusters.
- Les autorisations Amazon DocumentDB sont utilisées lors des appels d'API pour valider les ressources transmises dans une demande. Ils sont nécessaires pour qu'Amazon DocumentDB puisse utiliser la clé transmise avec le cluster Amazon DocumentDB.

- Les CloudWatch journaux sont nécessaires pour qu'Amazon DocumentDB puisse garantir que les destinations de livraison des journaux sont accessibles et qu'ils sont valides pour l'utilisation des journaux des courtiers.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "rds:AddRoleToDBCluster",
                "rds:AddSourceIdentifierToSubscription",
                "rds:AddTagsToResource",
                "rds:ApplyPendingMaintenanceAction",
                "rds:CopyDBClusterParameterGroup",
                "rds:CopyDBClusterSnapshot",
                "rds:CopyDBParameterGroup",
                "rds:CreateDBCluster",
                "rds:CreateDBClusterParameterGroup",
                "rds:CreateDBClusterSnapshot",
                "rds:CreateDBInstance",
                "rds:CreateDBParameterGroup",
                "rds:CreateDBSubnetGroup",
                "rds:CreateEventSubscription",
                "rds:DeleteDBCluster",
                "rds:DeleteDBClusterParameterGroup",
                "rds:DeleteDBClusterSnapshot",
                "rds:DeleteDBInstance",
                "rds:DeleteDBParameterGroup",
                "rds:DeleteDBSubnetGroup",
                "rds:DeleteEventSubscription",
                "rds:DescribeAccountAttributes",
                "rds:DescribeCertificates",
                "rds:DescribeDBClusterParameterGroups",
                "rds:DescribeDBClusterParameters",
                "rds:DescribeDBClusterSnapshotAttributes",
                "rds:DescribeDBClusterSnapshots",
                "rds:DescribeDBClusters",
                "rds:DescribeDBEngineVersions",
                "rds:DescribeDBInstances",
                "rds:DescribeDBLogFiles",
                "rds:DescribeDBParameterGroups",
                "rds:DescribeDBParameters",
```

```
                "rds:DescribeDBSecurityGroups",
                "rds:DescribeDBSubnetGroups",
                "rds:DescribeEngineDefaultClusterParameters",
                "rds:DescribeEngineDefaultParameters",
                "rds:DescribeEventCategories",
                "rds:DescribeEventSubscriptions",
                "rds:DescribeEvents",
                "rds:DescribeOptionGroups",
                "rds:DescribeOrderableDBInstanceOptions",
                "rds:DescribePendingMaintenanceActions",
                "rds:DescribeValidDBInstanceModifications",
                "rds:DownloadDBLogFilePortion",
                "rds:FailoverDBCluster",
                "rds:ListTagsForResource",
                "rds:ModifyDBCluster",
                "rds:ModifyDBClusterParameterGroup",
                "rds:ModifyDBClusterSnapshotAttribute",
                "rds:ModifyDBInstance",
                "rds:ModifyDBParameterGroup",
                "rds:ModifyDBSubnetGroup",
                "rds:ModifyEventSubscription",
                "rds:PromoteReadReplicaDBCluster",
                "rds:RebootDBInstance",
                "rds:RemoveRoleFromDBCluster",
                "rds:RemoveSourceIdentifierFromSubscription",
                "rds:RemoveTagsFromResource",
                "rds:ResetDBClusterParameterGroup",
                "rds:ResetDBParameterGroup",
                "rds:RestoreDBClusterFromSnapshot",
                "rds:RestoreDBClusterToPointInTime"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Action": [
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
```

```
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcs",
                "kms:ListAliases",
                "kms:ListKeyPolicies",
                "kms:ListKeys",
                "kms:ListRetirableGrants",
                "logs:DescribeLogStreams",
                "logs:GetLogEvents",
                "sns:ListSubscriptions",
                "sns:ListTopics",
                "sns:Publish"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Action": "iam:CreateServiceLinkedRole",
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
            "Condition": {
                "StringLike": {
                    "iam:AWS ServiceName": "rds.amazonaws.com"
                }
            }
        }
    ]
}
```

AmazonDocDB ReadOnlyAccess

Cette politique accorde des autorisations en lecture seule qui permettent aux utilisateurs de consulter les informations dans Amazon DocumentDB. Les directeurs auxquels cette politique est attachée ne peuvent effectuer aucune mise à jour ou supprimer des ressources existantes, ni créer de nouvelles ressources Amazon DocumentDB. Par exemple, les principaux disposant de ces autorisations peuvent consulter la liste des clusters et des configurations associés à leur compte, mais ne peuvent pas modifier la configuration ou les paramètres des clusters. Les autorisations définies dans cette politique sont regroupées comme suit :

- Les autorisations Amazon DocumentDB vous permettent de répertorier les ressources Amazon DocumentDB, de les décrire et d'obtenir des informations les concernant.
- Les autorisations Amazon EC2 sont utilisées pour décrire le VPC Amazon, les sous-réseaux, les groupes de sécurité et les ENI associés à un cluster.
- Une autorisation Amazon DocumentDB est utilisée pour décrire la clé associée au cluster.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "rds:DescribeAccountAttributes",
                "rds:DescribeCertificates",
                "rds:DescribeDBClusterParameterGroups",
                "rds:DescribeDBClusterParameters",
                "rds:DescribeDBClusterSnapshotAttributes",
                "rds:DescribeDBClusterSnapshots",
                "rds:DescribeDBClusters",
                "rds:DescribeDBEngineVersions",
                "rds:DescribeDBInstances",
                "rds:DescribeDBLogFiles",
                "rds:DescribeDBParameterGroups",
                "rds:DescribeDBParameters",
                "rds:DescribeDBSubnetGroups",
                "rds:DescribeEventCategories",
                "rds:DescribeEventSubscriptions",
                "rds:DescribeEvents",
                "rds:DescribeOrderableDBInstanceOptions",
                "rds:DescribePendingMaintenanceActions",
                "rds:DownloadDBLogFilePortion",
                "rds:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics"
            ],
            "Effect": "Allow",
```

```
            "Resource": "*"
        },
        {
            "Action": [
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcs"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "kms:ListKeys",
                "kms:ListRetirableGrants",
                "kms:ListAliases",
                "kms:ListKeyPolicies"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "logs:DescribeLogStreams",
                "logs:GetLogEvents"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/rds/*:log-stream:*",
                "arn:aws:logs:*:*:log-group:/aws/docdb/*:log-stream:*"
            ]
        }
    ]
}
```

AmazonDocDB ConsoleFullAccess

Accorde un accès complet à la gestion des ressources Amazon DocumentDB à l' AWS Management Console aide des méthodes suivantes :

- Les autorisations Amazon DocumentDB permettant d'autoriser toutes les actions de cluster Amazon DocumentDB et Amazon DocumentDB.
- Certaines des autorisations Amazon EC2 définies dans cette politique sont requises pour valider les ressources transmises dans le cadre d'une demande d'API. Cela permet de s'assurer qu'Amazon DocumentDB est en mesure d'utiliser correctement les ressources pour approvisionner et gérer le cluster. Les autres autorisations Amazon EC2 de cette politique permettent à Amazon DocumentDB de AWS créer les ressources nécessaires pour vous permettre de vous connecter à vos clusters tels que VPCEndpoint.
- AWS KMS les autorisations sont utilisées lors des appels d'API AWS KMS pour valider les ressources transmises dans une demande. Ils sont nécessaires pour qu'Amazon DocumentDB puisse utiliser la clé transmise pour chiffrer et déchiffrer les données au repos avec le cluster élastique Amazon DocumentDB.
- Les CloudWatch journaux sont nécessaires pour qu'Amazon DocumentDB puisse garantir que les destinations de livraison des journaux sont accessibles et qu'ils sont valides pour l'audit et le profilage de l'utilisation des journaux.
- Les autorisations de Secrets Manager sont requises pour valider un secret donné et l'utiliser pour configurer l'utilisateur administrateur pour les clusters élastiques Amazon DocumentDB.
- Les autorisations Amazon RDS sont requises pour les actions de gestion du cluster Amazon DocumentDB. Pour certaines fonctionnalités de gestion, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon RDS.
- Les autorisations SNS permettent aux principaux d'accéder à des abonnements et à des rubriques Amazon Simple Notification Service (Amazon SNS), ainsi que de publier des messages Amazon SNS.
- Les autorisations IAM sont requises pour créer les rôles liés au service requis pour la publication des métriques et des journaux.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DocdbSids",
            "Effect": "Allow",
            "Action": [
                "docdb-elastic:CreateCluster",
                "docdb-elastic:UpdateCluster",
                "docdb-elastic:GetCluster",
```

```
                "docdb-elastic:DeleteCluster",
                "docdb-elastic:ListClusters",
                "docdb-elastic:CreateClusterSnapshot",
                "docdb-elastic:GetClusterSnapshot",
                "docdb-elastic:DeleteClusterSnapshot",
                "docdb-elastic:ListClusterSnapshots",
                "docdb-elastic:RestoreClusterFromSnapshot",
                "docdb-elastic:TagResource",
                "docdb-elastic:UntagResource",
                "docdb-elastic:ListTagsForResource",
                "docdb-elastic:CopyClusterSnapshot",
                "docdb-elastic:StartCluster",
                "docdb-elastic:StopCluster",
                "rds:AddRoleToDBCluster",
                "rds:AddSourceIdentifierToSubscription",
                "rds:AddTagsToResource",
                "rds:ApplyPendingMaintenanceAction",
                "rds:CopyDBClusterParameterGroup",
                "rds:CopyDBClusterSnapshot",
                "rds:CopyDBParameterGroup",
                "rds:CreateDBCluster",
                "rds:CreateDBClusterParameterGroup",
                "rds:CreateDBClusterSnapshot",
                "rds:CreateDBInstance",
                "rds:CreateDBParameterGroup",
                "rds:CreateDBSubnetGroup",
                "rds:CreateEventSubscription",
                "rds:CreateGlobalCluster",
                "rds:DeleteDBCluster",
                "rds:DeleteDBClusterParameterGroup",
                "rds:DeleteDBClusterSnapshot",
                "rds:DeleteDBInstance",
                "rds:DeleteDBParameterGroup",
                "rds:DeleteDBSubnetGroup",
                "rds:DeleteEventSubscription",
                "rds:DeleteGlobalCluster",
                "rds:DescribeAccountAttributes",
                "rds:DescribeCertificates",
                "rds:DescribeDBClusterParameterGroups",
                "rds:DescribeDBClusterParameters",
                "rds:DescribeDBClusterSnapshotAttributes",
                "rds:DescribeDBClusterSnapshots",
                "rds:DescribeDBClusters",
                "rds:DescribeDBEngineVersions",
```

```
                "rds:DescribeDBInstances",
                "rds:DescribeDBLogFiles",
                "rds:DescribeDBParameterGroups",
                "rds:DescribeDBParameters",
                "rds:DescribeDBSecurityGroups",
                "rds:DescribeDBSubnetGroups",
                "rds:DescribeEngineDefaultClusterParameters",
                "rds:DescribeEngineDefaultParameters",
                "rds:DescribeEventCategories",
                "rds:DescribeEventSubscriptions",
                "rds:DescribeEvents",
                "rds:DescribeGlobalClusters",
                "rds:DescribeOptionGroups",
                "rds:DescribeOrderableDBInstanceOptions",
                "rds:DescribePendingMaintenanceActions",
                "rds:DescribeValidDBInstanceModifications",
                "rds:DownloadDBLogFilePortion",
                "rds:FailoverDBCluster",
                "rds:ListTagsForResource",
                "rds:ModifyDBCluster",
                "rds:ModifyDBClusterParameterGroup",
                "rds:ModifyDBClusterSnapshotAttribute",
                "rds:ModifyDBInstance",
                "rds:ModifyDBParameterGroup",
                "rds:ModifyDBSubnetGroup",
                "rds:ModifyEventSubscription",
                "rds:ModifyGlobalCluster",
                "rds:PromoteReadReplicaDBCluster",
                "rds:RebootDBInstance",
                "rds:RemoveFromGlobalCluster",
                "rds:RemoveRoleFromDBCluster",
                "rds:RemoveSourceIdentifierFromSubscription",
                "rds:RemoveTagsFromResource",
                "rds:ResetDBClusterParameterGroup",
                "rds:ResetDBParameterGroup",
                "rds:RestoreDBClusterFromSnapshot",
                "rds:RestoreDBClusterToPointInTime"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "DependencySids",
```

```
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "ec2:AllocateAddress",
                "ec2:AssignIpv6Addresses",
                "ec2:AssignPrivateIpAddresses",
                "ec2:AssociateAddress",
                "ec2:AssociateRouteTable",
                "ec2:AssociateSubnetCidrBlock",
                "ec2:AssociateVpcCidrBlock",
                "ec2:AttachInternetGateway",
                "ec2:AttachNetworkInterface",
                "ec2:CreateCustomerGateway",
                "ec2:CreateDefaultSubnet",
                "ec2:CreateDefaultVpc",
                "ec2:CreateInternetGateway",
                "ec2:CreateNatGateway",
                "ec2:CreateNetworkInterface",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSecurityGroup",
                "ec2:CreateSubnet",
                "ec2:CreateVpc",
                "ec2:CreateVpcEndpoint",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAddresses",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeCustomerGateways",
                "ec2:DescribeInstances",
                "ec2:DescribeNatGateways",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribePrefixLists",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroupReferences",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeVpcs",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:ModifySubnetAttribute",
```

```
                "ec2:ModifyVpcAttribute",
                "ec2:ModifyVpcEndpoint",
                "kms:DescribeKey",
                "kms:ListAliases",
                "kms:ListKeyPolicies",
                "kms:ListKeys",
                "kms:ListRetirableGrants",
                "logs:DescribeLogStreams",
                "logs:GetLogEvents",
                "sns:ListSubscriptions",
                "sns:ListTopics",
                "sns:Publish"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "DocdbSLRSid",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "rds.amazonaws.com"
                }
            }
        },
        {
            "Sid": "DocdbElasticSLRSid",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/docdb-
elastic.amazonaws.com/AWSServiceRoleForDocDB-Elastic",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "docdb-elastic.amazonaws.com"
                }
            }
        }
    ]
}
```

AmazonDocDB ElasticReadOnlyAccess

Cette politique accorde des autorisations en lecture seule qui permettent aux utilisateurs de consulter les informations relatives aux clusters élastiques dans Amazon DocumentDB. Les directeurs auxquels cette politique est attachée ne peuvent effectuer aucune mise à jour ou supprimer des ressources existantes, ni créer de nouvelles ressources Amazon DocumentDB. Par exemple, les principaux disposant de ces autorisations peuvent consulter la liste des clusters et des configurations associés à leur compte, mais ne peuvent pas modifier la configuration ou les paramètres des clusters. Les autorisations définies dans cette politique sont regroupées comme suit :

- Les autorisations du cluster élastique Amazon DocumentDB vous permettent de répertorier les ressources du cluster élastique Amazon DocumentDB, de les décrire et d'obtenir des informations à leur sujet.
- CloudWatch les autorisations sont utilisées pour vérifier les métriques de service.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "docdb-elastic:ListClusters",
            "docdb-elastic:GetCluster",
            "docdb-elastic:ListClusterSnapshots",
            "docdb-elastic:GetClusterSnapshot",
            "docdb-elastic:ListTagsForResource"
         ],
         "Resource": "*"
      },
      {
         "Effect": "Allow",
         "Action": [
            "cloudwatch:GetMetricData",
            "cloudwatch:ListMetrics",
            "cloudwatch:GetMetricStatistics"
         ],
         "Resource": "*"
      }
   ]
}
```

AmazonDocDB ElasticFullAccess

Cette politique accorde des autorisations administratives qui permettent un accès complet principal à toutes les actions Amazon DocumentDB pour le cluster élastique Amazon DocumentDB.

Cette politique utilise des AWS balises (https://docs.aws.amazon.com/tag-editor/latest/userguide/tagging.html) dans des conditions permettant de définir l'accès aux ressources. Si vous utilisez un secret, il doit être étiqueté avec une clé de balise `DocDBElasticFullAccess` et une valeur de balise. Si vous utilisez une clé gérée par le client, elle doit être étiquetée avec une clé de balise `DocDBElasticFullAccess` et une valeur de balise.

Les autorisations définies dans cette politique sont regroupées comme suit :

- Les autorisations du cluster élastique Amazon DocumentDB autorisent toutes les actions Amazon DocumentDB.
- Certaines des autorisations Amazon EC2 définies dans cette politique sont requises pour valider les ressources transmises dans le cadre d'une demande d'API. Cela permet de s'assurer qu'Amazon DocumentDB est en mesure d'utiliser correctement les ressources pour approvisionner et gérer le cluster. Les autres autorisations Amazon EC2 de cette politique permettent à Amazon DocumentDB de AWS créer les ressources nécessaires pour vous permettre de vous connecter à vos clusters comme un point de terminaison VPC.
- AWS KMS des autorisations sont requises pour qu'Amazon DocumentDB puisse utiliser la clé transmise pour chiffrer et déchiffrer les données au repos au sein du cluster élastique Amazon DocumentDB.

  > **Note**
  >
  > La clé gérée par le client doit comporter une étiquette avec clé `DocDBElasticFullAccess` et une valeur de balise.

- SecretsManager des autorisations sont requises pour valider le secret donné et l'utiliser pour configurer l'utilisateur administrateur pour les clusters élastiques Amazon DocumentDB.

  > **Note**
  >
  > Le secret utilisé doit avoir une balise avec clé `DocDBElasticFullAccess` et une valeur de balise.

- Les autorisations IAM sont requises pour créer les rôles liés au service requis pour la publication des métriques et des journaux.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DocdbElasticSid",
            "Effect": "Allow",
            "Action": [
                "docdb-elastic:CreateCluster",
                "docdb-elastic:UpdateCluster",
                "docdb-elastic:GetCluster",
                "docdb-elastic:DeleteCluster",
                "docdb-elastic:ListClusters",
                "docdb-elastic:CreateClusterSnapshot",
                "docdb-elastic:GetClusterSnapshot",
                "docdb-elastic:DeleteClusterSnapshot",
                "docdb-elastic:ListClusterSnapshots",
                "docdb-elastic:RestoreClusterFromSnapshot",
                "docdb-elastic:TagResource",
                "docdb-elastic:UntagResource",
                "docdb-elastic:ListTagsForResource",
                "docdb-elastic:CopyClusterSnapshot",
                "docdb-elastic:StartCluster",
                "docdb-elastic:StopCluster"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "EC2Sid",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVpcEndpoint",
                "ec2:DescribeVpcEndpoints",
                "ec2:DeleteVpcEndpoints",
                "ec2:ModifyVpcEndpoint",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
```

```
                "ec2:DescribeVpcs",
                "ec2:DescribeAvailabilityZones",
                "secretsmanager:ListSecrets"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "docdb-elastic.amazonaws.com"
                }
            }
        },
        {
            "Sid": "KMSSid",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "docdb-elastic.*.amazonaws.com"
                    ],
                    "aws:ResourceTag/DocDBElasticFullAccess": "*"
                }
            }
        },
        {
            "Sid": "KMSGrantSid",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/DocDBElasticFullAccess": "*",
                    "kms:ViaService": [
                        "docdb-elastic.*.amazonaws.com"
                    ]
```

```
                },
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
            }
        },
        {
            "Sid": "SecretManagerSid",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:ListSecretVersionIds",
                "secretsmanager:DescribeSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:GetResourcePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "secretsmanager:ResourceTag/DocDBElasticFullAccess": "*"
                },
                "StringEquals": {
                    "aws:CalledViaFirst": "docdb-elastic.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudwatchSid",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:ListMetrics",
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "SLRSid",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/docdb-
elastic.amazonaws.com/AWSServiceRoleForDocDB-Elastic",
            "Condition": {
```

```
                "StringLike": {
                    "iam:AWSServiceName": "docdb-elastic.amazonaws.com"
                }
            }
        }
    ]
}
```

AmazonDocDB- ElasticServiceRolePolicy

Vous ne pouvez pas vous attacher `AmazonDocDBElasticServiceRolePolicy` à vos AWS Identity and Access Management entités. Cette politique est associée à un rôle lié à un service qui permet à Amazon DocumentDB d'effectuer des actions en votre nom. Pour plus d'informations, consultez Rôles liés aux services dans les clusters élastiques.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "AWS/DocDB-Elastic"
                    ]
                }
            }
        }
    ]
}
```

Amazon DocumentDB met à jour les politiques gérées AWS

Modification	Description	Date
AmazonDocDB ElasticFullAccess, AmazonDocDB ConsoleFullAccess - Modifier	Les politiques ont été mises à jour pour ajouter des actions de démarrage/arrêt du cluster et de copie des instantanés du cluster.	21/02/2024
AmazonDocDB ElasticReadOnlyAccess, AmazonDocDB ElasticFullAccess - Modifier	Politiques mises à jour pour ajouter des `cloudwatch:GetMetricData` actions.	21/06/2023
AmazonDocDB ElasticReadOnlyAccess : nouvelle politique	Nouvelle politique gérée pour les clusters élastiques Amazon DocumentDB	08/06/2023
AmazonDocDB ElasticFullAccess : nouvelle politique	Nouvelle politique gérée pour les clusters élastiques Amazon DocumentDB	05/06/2023
AmazonDocDB- ElasticServiceRolePolicy : nouvelle politique	Amazon DocumentDB crée un nouveau rôle lié au service AWS ServiceRoleForDoc DB-Elastic pour les clusters élastiques Amazon DocumentDB	30/11/2022
AmazonDocDB ConsoleFullAccess- Changement	Politique mise à jour pour ajouter les autorisations de cluster globales et élastiques d'Amazon DocumentDB	30/11/2022
AmazonDocDB ConsoleFullAccess,AmazonDocDB FullAccess, AmazonDocDB	Lancement de service	19/01/2017

Modification	Description	Date
ReadOnlyAccess - Nouvelle politique		

Autorisations d'API Amazon DocumentDB : référence des actions, des ressources et des conditions

Utilisez les sections suivantes comme référence lorsque vous configurez Utilisation de politiques basées sur l'identité (politiques IAM) pour Amazon DocumentDB et rédigez des politiques d'autorisation que vous pouvez associer à une identité IAM (politiques basées sur l'identité).

La liste suivante répertorie chaque opération d'API Amazon DocumentDB. La liste comprend les actions correspondantes pour lesquelles vous pouvez accorder des autorisations pour effectuer l'action, la AWS ressource pour laquelle vous pouvez accorder les autorisations et les clés de condition que vous pouvez inclure pour un contrôle d'accès précis. Vous spécifiez les actions dans le champ `Action` de la politique, la valeur de ressource dans le champ `Resource` de la politique, et les conditions dans le champ `Condition` de la politique. Pour plus d'informations sur les conditions, consultez Spécification de conditions dans une politique.

Vous pouvez utiliser des clés AWS de condition larges dans vos politiques Amazon DocumentDB pour exprimer des conditions. Pour obtenir la liste complète des touches AWS-wide, consultez la section Clés disponibles dans le guide de l'utilisateur IAM.

Vous pouvez tester les politiques IAM à l'aide du simulateur de politiques IAM. Il fournit automatiquement une liste des ressources et des paramètres requis pour chaque AWS action, y compris les actions Amazon DocumentDB. Le simulateur de politique IAM détermine les autorisations requises pour chacune des actions que vous spécifiez. Pour plus d'informations sur le simulateur de politique IAM, voir Tester les politiques IAM avec le simulateur de politique IAM dans le guide de l'utilisateur IAM.

> **Note**
>
> Pour indiquer une action, utilisez le préfixe `rds:` suivi du nom de l'opération d'API (par exemple, `rds:CreateDBInstance`).

La liste suivante répertorie les opérations d'API Amazon RDS ainsi que leurs actions, ressources et clés de condition associées.

Rubriques

- Actions Amazon DocumentDB qui prennent en charge les autorisations au niveau des ressources
- Actions Amazon DocumentDB qui ne prennent pas en charge les autorisations au niveau des ressources

Actions Amazon DocumentDB qui prennent en charge les autorisations au niveau des ressources

Les autorisations au niveau des ressources permettent de spécifier les ressources sur lesquelles les utilisateurs sont autorisés à effectuer des actions. Amazon DocumentDB prend partiellement en charge les autorisations au niveau des ressources. Cela signifie que pour certaines actions Amazon DocumentDB, vous pouvez contrôler le moment où les utilisateurs sont autorisés à utiliser ces actions en fonction des conditions qui doivent être remplies ou des ressources spécifiques que les utilisateurs sont autorisés à utiliser. Par exemple, vous pouvez accorder aux utilisateurs l'autorisation de modifier uniquement des instances spécifiques.

La liste suivante répertorie les opérations de l'API Amazon DocumentDB ainsi que leurs actions, ressources et clés de condition associées.

> **Note**
>
> Pour certaines fonctionnalités de gestion, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon RDS. Pour plus d'actions et d'autorisations Amazon DocumentDB, reportez-vous à la section Actions, ressources et clés de condition pour Amazon RDS dans la référence d'autorisation de service.

Opérations et actions de l'API Amazon DocumentDB	Ressources	Clés de condition
AddTagsToResource	Instance	`rds:db-tag`

Opérations et actions de l'API Amazon DocumentDB	Ressources	Clés de condition
`rds:AddTagsToResource`	`arn:aws:rds:` *region*`:`*account-id*`:db:`*db-instance-name*	
	Groupe de sous-réseaux `arn:aws:rds:` *region*`:`*account-id*`:subgrp:`*subnet-group-name*	`rds:subgrp-tag`
ApplyPendingMaintenanceAction `rds:ApplyPendingMaintenanceAction`	Instance `arn:aws:rds:` *region*`:`*account-id*`:db:`*db-instance-name*	`rds:db-tag`
Copier DB ClusterSnapshot `rds:CopyDBClusterSnapshot`	Instantané du cluster `arn:aws:rds:` *region*`:`*account-id*`:cluster-snapshot:`*cluster-snapshot-name*	`rds:cluster-snapshot-tag`
CreateDBCluster `rds:CreateDBCluster`	Cluster `arn:aws:rds:` *region*`:`*account-id*`:cluster:` *db-cluster-name*	`rds:cluster-tag`
	Groupe de paramètres du cluster `arn:aws:rds:` *region*`:`*account-id*`:cluster-pg:` *cluster-parameter-group-name*	`rds:cluster-pg-tag`

Opérations et actions de l'API Amazon DocumentDB	Ressources	Clés de condition
	Groupe de sous-réseaux arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
Créer une base de données ClusterParameterGroup rds:CreateDBClusterParameterGroup	Groupe de paramètres du cluster arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag
Créer une base de données ClusterSnapshot rds:CreateDBClusterSnapshot	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-name*	rds:cluster-tag
	Instantané du cluster arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
CreateDBInstance rds:CreateDBInstance	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:DatabaseClass rds:db-tag

Opérations et actions de l'API Amazon DocumentDB	Ressources	Clés de condition
	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-name*	rds:cluster-tag
Créer une base de données SubnetGroup rds:CreateDBSubnetGroup	Groupe de sous-réseaux arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
DeleteDBInstance rds:DeleteDBInstance	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
Supprimer B SubnetGroup rds:DeleteDBSubnetGroup	Groupe de sous-réseaux arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
Décrit B ClusterParameterGroups rds:DescribeDBClusterParameterGroups	Groupe de paramètres du cluster arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag

Opérations et actions de l'API Amazon DocumentDB	Ressources	Clés de condition
Décrit B ClusterParameters rds:DescribeDBClusterParameters	Groupe de paramètres du cluster arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag
DescribeDBClusters rds:DescribeDBClusters	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-instance-name*	rds:cluster-tag
Décrit B ClusterSnapshotAttributes rds:DescribeDBClusterSnapshotAttributes	Instantané du cluster arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
Décrit B SubnetGroups rds:DescribeDBSubnetGroups	Groupe de sous-réseaux arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
DescribePendingMaintenanceActions rds:DescribePendingMaintenanceActions	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:DatabaseClass rds:db-tag

Opérations et actions de l'API Amazon DocumentDB	Ressources	Clés de condition
FailoverDBCluster rds:FailoverDBCluster	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-instance-name*	rds:cluster-tag
ListTagsForResource rds:ListTagsForResource	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
	Groupe de sous-réseaux arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
ModifyDBCluster rds:ModifyDBCluster	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-name*	rds:cluster-tag
	Groupe de paramètres du cluster arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag

Opérations et actions de l'API Amazon DocumentDB	Ressources	Clés de condition
Modifier la base de données ClusterParameterGroup rds:ModifyDBClusterParameterGroup	Groupe de paramètres du cluster arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag
Modifier la base de données ClusterSnapshotAttribute rds:ModifyDBClusterSnapshotAttribute	Instantané du cluster arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
ModifyDBInstance rds:ModifyDBInstance	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:DatabaseClass rds:db-tag
RebootDBInstance rds:RebootDBInstance	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag

Opérations et actions de l'API Amazon DocumentDB	Ressources	Clés de condition
RemoveTagsFromResource rds:RemoveTagsFromResource	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
	Groupe de sous-réseaux arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
Réinitialiser DB ClusterParameterGroup rds:ResetDBClusterParameterGroup	Groupe de paramètres du cluster arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag
Restaurer la base de données ClusterFromSnapshot rds:RestoreDBClusterFromSnapshot	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-instance-name*	rds:cluster-tag
	Instantané du cluster arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag

Opérations et actions de l'API Amazon DocumentDB	Ressources	Clés de condition
Restaurer la base de données ClusterTo PointInTime `rds:RestoreDBClusterToPointInTime`	Cluster `arn:aws:rds:` *`region`*`:`*`account-id`*` :cluster:` *`db-cluster-instance-name`*	`rds:cluster-tag`
	Groupe de sous-réseaux `arn:aws:rds:` *`region`*`:`*`account-id`*` :subgrp:`*`subnet-group-name`*	`rds:subgrp-tag`

Actions Amazon DocumentDB qui ne prennent pas en charge les autorisations au niveau des ressources

Vous pouvez utiliser toutes les actions Amazon DocumentDB dans une politique IAM pour accorder ou refuser aux utilisateurs l'autorisation d'utiliser cette action. Cependant, toutes les actions Amazon DocumentDB ne prennent pas en charge les autorisations au niveau des ressources, qui vous permettent de spécifier les ressources sur lesquelles une action peut être effectuée. Les actions d'API Amazon DocumentDB suivantes ne prennent actuellement pas en charge les autorisations au niveau des ressources. Par conséquent, pour utiliser ces actions dans une politique IAM, vous devez autoriser les utilisateurs à utiliser toutes les ressources nécessaires à l'action en utilisant un * caractère générique pour l'`Resource`élément de votre déclaration.

- `rds:DescribeDBClusterSnapshots`
- `rds:DescribeDBInstances`

Gestion des utilisateurs Amazon DocumentDB

Dans Amazon DocumentDB, les utilisateurs s'authentifient auprès d'un cluster en conjonction avec un mot de passe. Chaque cluster possède des informations de connexion principales qui sont établies lors de la création du cluster.

> **Note**
>
> Tous les nouveaux utilisateurs créés avant le 26 mars 2020 ont reçu les rôles `clusterAdmin`, `dbAdminAnyDatabase` et `readWriteAnyDatabase`. Il est recommandé de réévaluer tous les utilisateurs et de modifier les rôles si nécessaire pour appliquer le principe du moindre privilège à tous les utilisateurs de vos clusters.
> Pour plus d'informations, veuillez consulter Accès à la base de données à l'aide du contrôle d'accès basé sur les rôles.

Principal et **`serviceadmin`** utilisateur

Un cluster Amazon DocumentDB récemment créé compte deux utilisateurs : l'utilisateur principal et l'`serviceadmin`utilisateur.

L'utilisateur principal est un utilisateur privilégié unique qui peut effectuer des tâches administratives et créer des utilisateurs supplémentaires dotés de rôles. Lorsque vous vous connectez à un cluster Amazon DocumentDB pour la première fois, vous devez vous authentifier à l'aide des informations de connexion principales. L'utilisateur principal reçoit ces autorisations administratives pour un cluster Amazon DocumentDB lors de la création de ce cluster et se voit attribuer le rôle de`root`.

L'utilisateur `serviceadmin` est créé implicitement lors de la création du cluster. Chaque cluster Amazon DocumentDB possède un`serviceadmin` utilisateur qui permetAWS de gérer votre cluster. Vous ne pouvez pas vous connecter en tant que, supprimer, renommer, modifier le mot de passe ou modifier les autorisations pour `serviceadmin`. Toute tentative de ce type produit une erreur.

> **Note**
>
> Le principal et`serviceadmin` les utilisateurs d'un cluster Amazon DocumentDB ne peuvent pas être supprimés et le rôle`root` de l'utilisateur principal ne peut pas être révoqué.
> Si vous oubliez votre mot de passe d'utilisateur principal, vous pouvez le réinitialiser à l'aide de laAWS Management Console ou de laAWS CLI.

Création d'utilisateurs supplémentaires

Après vous être connecté en tant qu'utilisateur principal (ou tout autre utilisateur ayant ce rôle`createUser`), vous pouvez créer un nouvel utilisateur, comme indiqué ci-dessous.

```
db.createUser(
    {
        user: "sample-user-1",
        pwd: "password123",
        roles:
            [{"db":"admin", "role":"dbAdminAnyDatabase" }]
    }
)
```

Pour afficher les détails de l'utilisateur, vous pouvez utiliser la commande `show users` comme suit. Vous pouvez également supprimer des utilisateurs avec la commande `dropUser`. Pour plus d'informations, veuillez consulter Commandes courantes.

```
show users
{
    "_id" : "serviceadmin",
    "user" : "serviceadmin",
    "db" : "admin",
    "roles" : [
     {
            "role" : "root",
            "db" : "admin"
        }
    ]
},

{
    "_id" : "myPrimaryUser",
    "user" : "myPrimaryUser",
    "db" : "admin",
    "roles" : [
     {
            "role" : "root",
            "db" : "admin"
        }
    ]
},

{
    "_id" : "sample-user-1",
    "user" : "sample-user-1",
```

```
    "db" : "admin",
    "roles" : [
     {
            "role" : "dbAdminAnyDatabase",
            "db" : "admin"
     }
    ]
}
```

Dans l'exemple ci-dessus, le nouvel utilisateur `sample-user-1` est attribué à la base de données `admin`. C'est toujours le cas pour un nouvel utilisateur. Amazon DocumentDB n'a pas le concept d'authentification`authenticationDatabase` et, par conséquent, toutes les authentifications sont effectuées dans le contexte de la`admin` base de données.

Lors de la création d'utilisateurs, si vous omettez le`db` champ lorsque vous spécifiez le rôle, Amazon DocumentDB attribuera implicitement le rôle à la base de données dans laquelle la connexion est établie. Par exemple, si votre connexion est émise sur la base de données `sample-database` et que vous exécutez la commande suivante, l'utilisateur `sample-user-2` sera créé dans la base de données `admin` et aura des autorisations `readWrite` sur la base de données `sample-database`.

```
db.createUser(
    {
        user: "sample-user-2",
        pwd: "password123",
        roles:
            ["readWrite"]
    }
)
```

La création d'utilisateurs avec des rôles étendus à toutes les bases de données (par exemple, `readInAnyDatabase`) nécessite que vous soyez dans le contexte de la base de données `admin` lors de la création de l'utilisateur, ou que vous indiquiez explicitement la base de données pour le rôle lors de la création de l'utilisateur.

Pour changer le contexte de votre base de données, vous pouvez utiliser la commande suivante.

```
use admin
```

Pour en savoir plus sur le contrôle d'accès basé sur les rôles et l'application du principe du moindre privilège parmi les utilisateurs de votre cluster, veuillez consulter Accès à la base de données à l'aide du contrôle d'accès basé sur les rôles.

Rotation automatique des mots de passe pour Amazon DocumentDB

Avec AWS Secrets Manager, vous pouvez remplacer les informations d'identification codées en dur dans votre code (y compris les mots de passe) par un appel d'API à Secrets Manager pour récupérer le secret par programmation. Cela permet de garantir que le secret ne peut pas être mis en péril par une personne qui examine votre code, étant donné que le secret n'y figure pas. En outre, vous pouvez configurer Secrets Manager afin d'effectuer automatiquement une rotation du secret, selon une planification que vous spécifiez. Cela vous permet de remplacer les secrets à long terme par ceux à court terme, ce qui réduit considérablement le risque de mise en péril.

Secrets Manager vous permet d'assurer la rotation automatique des mots de passe Amazon DocumentDBAWS Lambda

PourAWS Secrets Manager de informations sur Amazon DocumentDB

- Blog : Comment alterner les informations d'identification Amazon DocumentDB et Amazon Redshift dansAWS Secrets Manager
- Qu'est-ce queAWS Secrets Manager ?
- Rotation des secrets pour Amazon DocumentDB

Accès à la base de données à l'aide du contrôle d'accès basé sur les rôles

Vous pouvez restreindre l'accès aux actions que les utilisateurs peuvent effectuer sur les bases de données à l'aide du contrôle d'accès basé sur les rôles (RBAC) dans Amazon DocumentDB (avec compatibilité avec MongoDB). Le RBAC fonctionne en accordant un ou plusieurs rôles à un utilisateur. Ces rôles déterminent les opérations qu'un utilisateur peut effectuer sur les ressources de base de données. Amazon DocumentDB prend actuellement en charge à la fois les rôles intégrés définis au niveau de la base de données, tels que,,`read`, `readWrite` `readAnyDatabase``clusterAdmin`, et les rôles définis par l'utilisateur pouvant être étendus à des actions spécifiques, ainsi que des ressources granulaires telles que des collections en fonction de vos besoins.

Les cas d'utilisation courants du RBAC incluent l'application des privilèges minimaux en créant des utilisateurs ayant un accès en lecture seule aux bases de données ou aux collections d'un cluster, et les conceptions d'applications multi-locataires qui permettent à un seul utilisateur d'accéder à une base de données ou à une collection donnée dans un cluster.

> **Note**
>
> Tous les nouveaux utilisateurs créés avant le 26 mars 2020 ont reçu les rôles `clusterAdmin`, `dbAdminAnyDatabase` et `readWriteAnyDatabase`. Il est recommandé de réévaluer tous les utilisateurs existants et de modifier les rôles si nécessaire pour appliquer le principe du moindre privilège pour vos clusters.

Rubriques

Concepts RBAC

Voici des termes et concepts importants liés au contrôle d'accès basé sur les rôles. Pour plus d'informations sur les utilisateurs d'Amazon DocumentDB, consultez. Gestion des utilisateurs Amazon DocumentDB

- Utilisateur : entité individuelle capable de s'authentifier auprès de la base de données et d'effectuer des opérations.
- Mot de passe : secret utilisé pour authentifier l'utilisateur.
- Rôle : autorise un utilisateur à effectuer des actions sur une ou plusieurs bases de données.
- Base de données d'administration : base de données dans laquelle les utilisateurs sont enregistrés et autorisés.

- Database (**db**) — L'espace de noms au sein des clusters qui contient des collections pour le stockage de documents.

La commande suivante permet de créer un utilisateur nommé `sample-user`.

```
db.createUser({user: "sample-user", pwd: "abc123", roles: [{role: "read", db: "sample-database"}]})
```

Dans cet exemple :

- `user: "sample-user"`— Indique le nom d'utilisateur.
- `pwd: "abc123"`— Indique le mot de passe de l'utilisateur.
- `role: "read", "db: "sample-database"`— Indique que l'utilisateur `sample-user` disposera d'autorisations de lecture`sample-database`.

L'exemple suivant présente la sortie après l'obtention de l'utilisateur `sample-user` avec `db.getUser(sample-user)`. Dans cet exemple, l'utilisateur `sample-user` réside dans la base de données `admin` mais possède le rôle de lecture pour la base de données `sample-database`.

Lorsque vous créez des utilisateurs, si vous omettez le db champ lorsque vous spécifiez le rôle, Amazon DocumentDB attribuera implicitement le rôle à la base de données dans laquelle la

connexion est établie. Par exemple, si votre connexion est émise sur la base de données `sample-database` et que vous exécutez la commande suivante, l'utilisateur `sample-user` sera créé dans la base de données `admin` et aura des autorisations `readWrite` sur la base de données `sample-database`.

```
db.createUser({user: "sample-user", pwd: "abc123", roles: ["readWrite"]})
```

Le résultat de cette opération ressemble à ceci.

```
{
   "user":"sample-user",
   "roles":[
      {
         "db":"sample-database",
         "role":"readWrite"
      }
   ]
}
```

La création d'utilisateurs avec des rôles étendus à toutes les bases de données (par exemple, `readAnyDatabase`) nécessite que vous soyez dans le contexte de la base de données `admin` lors de la création de l'utilisateur, ou que vous indiquiez explicitement la base de données pour le rôle lors de la création de l'utilisateur. Pour émettre des commandes sur la base de données `admin`, vous pouvez utiliser la commande `use admin`. Pour plus d'informations, consultez Commandes courantes.

Commencer à utiliser les rôles intégrés au RBAC

Pour vous aider à démarrer avec le contrôle d'accès basé sur les rôles, cette section vous guide à travers un exemple de scénario d'application du principe du moindre privilège en créant des rôles pour trois utilisateurs aux fonctions de travail différentes.

- `user1` est un nouveau responsable qui doit pouvoir afficher et accéder à toutes les bases de données d'un cluster.
- `user2` est un nouvel employé qui a besoin d'accéder à une seule base de données, `sample-database-1`, dans ce même cluster.
- `user3` est un employé existant qui doit afficher et accéder à une base de données différente, `sample-database-2`, à laquelle il n'avait pas accès auparavant, dans le même cluster.

Plus tard, `user1` et `user2` quitteront la société et leur accès devra donc être révoqué.

Pour créer des utilisateurs et accorder des rôles, l'utilisateur avec lequel vous vous authentifiez auprès du cluster doit avoir un rôle associé pouvant effectuer des actions pour `createUser` et `grantRole`. Par exemple, les rôles `admin` et `userAdminAnyDatabase` peuvent tous les deux accorder ces autorisations. Pour les actions par rôle, veuillez consulter Accès à la base de données à l'aide du contrôle d'accès basé sur les rôles.

> **Note**
>
> Dans Amazon DocumentDB, toutes les opérations relatives aux utilisateurs et aux rôles (par exemple,`create`,,`get`,`drop`, `grantrevoke`, etc.) sont implicitement effectuées dans la `admin` base de données, que vous émettiez ou non des commandes sur la base de données. `admin`

Tout d'abord, pour comprendre quels sont les utilisateurs et les rôles actuels dans le cluster, vous pouvez exécuter la commande `show users`, comme dans l'exemple suivant. Vous verrez deux utilisateurs, `serviceadmin` et l'utilisateur principal du cluster. Ces deux utilisateurs existent toujours et ne peuvent pas être supprimés. Pour plus d'informations, consultez Gestion des utilisateurs Amazon DocumentDB.

```
show users
```

Pour `user1`, créez un rôle avec un accès en lecture et en écriture sur toutes les bases de données de l'ensemble du cluster à l'aide de la commande suivante.

```
db.createUser({user: "user1", pwd: "abc123", roles: [{role: "readWriteAnyDatabase", db:
 "admin"}]})
```

Le résultat de cette opération ressemble à ceci.

```
{
   "user":"user1",
   "roles":[
      {
         "role":"readWriteAnyDatabase",
         "db":"admin"
      }
```

```
   ]
}
```

Pour `user2`, créez un rôle avec un accès en lecture seule sur la base de données `sample-database-1` à l'aide de la commande suivante.

```
db.createUser({user: "user2", pwd: "abc123", roles: [{role: "read", db: "sample-
database-1"}]})
```

Le résultat de cette opération ressemble à ceci.

```
{
   "user":"user2",
   "roles":[
      {
         "role":"read",
         "db":"sample-database-1"
      }
   ]
}
```

Pour simuler le scénario où `user3` est un utilisateur existant, créez d'abord l'utilisateur `user3`, puis affectez un nouveau rôle à `user3`.

```
db.createUser({user: "user3", pwd: "abc123", roles: [{role: "readWrite", db: "sample-
database-1"}]})
```

Le résultat de cette opération ressemble à ceci.

```
{
   "user":"user3",
   "roles":[
      {
         "role":"readWrite",
         "db":"sample-database-1"
      }
   ]
}
```

Maintenant que l'utilisateur `user3` a été créé, affectez à `user3` le rôle `read` sur `sample-database-2`.

```
db.grantRolesToUser("user3", [{role: "read", db: "sample-database-2"}])
```

Enfin, `user1` et `user2` quittent l'entreprise, leur accès au cluster doit donc être révoqué. Pour ce faire, vous pouvez supprimer les utilisateurs, comme suit.

```
db.dropUser("user1")
db.dropUser("user2")
```

Pour vous assurer que tous les utilisateurs disposent des rôles appropriés, vous pouvez répertorier tous les utilisateurs avec la commande suivante.

```
show users
```

Le résultat de cette opération ressemble à ceci.

```
{
   "_id":"serviceadmin",
   "user":"serviceadmin",
   "db":"admin",
   "roles":[
      {
         "db":"admin",
         "role":"root"
      }
   ]
}
{
   "_id":"master-user",
   "user":"master-user",
   "db":"admin",
   "roles":[
      {
         "db":"admin",
         "role":"root"
      }
   ]
}
{
   "_id":"user3",
   "user":"user3",
   "db":"admin",
```

```
   "roles":[
      {
         "db":"sample-database-2",
         "role":"read"
      },
      {
         "db":"sample-database-1",
         "role":"readWrite"
      }
   ]
}
```

Commencer à utiliser les rôles définis par l'utilisateur du RBAC

Pour vous aider à démarrer avec les rôles définis par l'utilisateur, cette section présente un exemple de scénario d'application du moindre privilège en créant des rôles pour trois utilisateurs ayant des fonctions différentes.

Dans cet exemple, les règles suivantes s'appliquent :

- `user1` est un nouveau responsable qui doit pouvoir afficher et accéder à toutes les bases de données d'un cluster.
- `user2`est un nouvel employé qui n'a besoin que de l'action « rechercher » sur une seule base de données`sample-database-1`, dans le même cluster.
- `user3`est un employé existant qui doit consulter et accéder à une collection spécifique, col2, dans une base de données différente, à `sample-database-2` laquelle il n'avait pas accès auparavant, dans le même cluster.
- Pour `user1`, créez un rôle avec un accès en lecture et en écriture sur toutes les bases de données de l'ensemble du cluster à l'aide de la commande suivante.

```
db.createUser(
 {
     user: "user1", pwd: "abc123",
     roles: [{role: "readWriteAnyDatabase", db: "admin"}]
 }
 )
```

Le résultat de cette opération ressemble à ceci.

```
{
   "user":"user1",
   "roles":[
      {
         "role":"readWriteAnyDatabase",
         "db":"admin"
      }
   ]
}
```

Pour`user2`, créez un rôle doté des privilèges « find » pour toutes les collections de la base de données à l'`sample-database-1`aide de la commande suivante. Notez que ce rôle garantit que les utilisateurs associés ne peuvent exécuter que des requêtes de recherche.

```
db.createRole(
{
    role: "findRole",
    privileges: [
    {
        resource: {db: "sample-database-1", collection: ""}, actions: ["find"]
    }],
    roles: []
}
)
```

Le résultat de cette opération ressemble à ceci.

```
{
   "role":"findRole",
   "privileges":[
      {
         "resource":{
            "db":"sample-database-1",
            "collection":""
         },
         "actions":[
            "find"
         ]
      }
   ],
   "roles":[
```

```
   ]
}
```

Créez ensuite l'utilisateur (`user2`) et associez le rôle récemment créé `findRole` à l'utilisateur.

```
db.createUser(
{
    user: "user2",
    pwd: "abc123",
    roles: []
})

 db.grantRolesToUser("user2",["findRole"])
```

Pour simuler le scénario d'un utilisateur existant, créez d'abord l'utilisateur`user3`, puis créez un nouveau rôle appelé CollectionRole que nous allons attribuer à l'étape suivante. `user3` `user3`

Vous pouvez désormais attribuer un nouveau rôle à`user3`. Ce nouveau rôle permettra d'insérer, de mettre `user3` à jour, de supprimer et de trouver l'accès à une collection spécifique dans `sample-database-2` laquelle col2 est installé.

```
db.createUser(
{
    user: "user3",
    pwd: "abc123",
    roles: []
})

db.createRole(
{
    role: "collectionRole",
    privileges: [
    {
        resource: {db: "sample-database-2", collection: "col2"}, actions: ["find",
 "update", "insert", "remove"]
    }],
    roles: []
}
)
```

Le résultat de cette opération ressemble à ceci.

```
{
   "role":"collectionRole",
   "privileges":[
      {
         "resource":{
            "db":"sample-database-2",
            "collection":"col2"
         },
         "actions":[
            "find",
            "update",
            "insert",
            "remove"
         ]
      }
   ],
   "roles":[

   ]
}
```

Maintenant que l'utilisateur `user3` a été créé, vous pouvez lui attribuer `user3` le rôle`collectionFind`.

```
db.grantRolesToUser("user3",["collectionRole"])
```

Enfin, `user1` et `user2` quittent l'entreprise, leur accès au cluster doit donc être révoqué. Pour ce faire, vous pouvez supprimer les utilisateurs, comme suit.

```
db.dropUser("user1")
db.dropUser("user2")
```

Pour vous assurer que tous les utilisateurs disposent des rôles appropriés, vous pouvez répertorier tous les utilisateurs avec la commande suivante.

```
show users
```

Le résultat de cette opération ressemble à ceci.

```
{
   "_id":"serviceadmin",
```

```
   "user":"serviceadmin",
   "db":"admin",
   "roles":[
      {
         "db":"admin",
         "role":"root"
      }
   ]
}
{
   "_id":"master-user",
   "user":"master-user",
   "db":"admin",
   "roles":[
      {
         "db":"admin",
         "role":"root"
      }
   ]
}
{
   "_id":"user3",
   "user":"user3",
   "db":"admin",
   "roles":[
      {
         "db":"admin",
         "role":"collectionRole"
      }
   ]
}
```

Connexion à Amazon DocumentDB en tant qu'utilisateur

Lorsque vous vous connectez à un cluster Amazon DocumentDB, vous vous connectez dans le contexte d'une base de données particulière. Par défaut, si vous ne spécifiez pas de base de données dans votre chaîne de connexion, vous êtes automatiquement connecté au cluster dans le contexte de la base de données `test`. Toutes les commandes de niveau collection comme `insert` et `find` sont émises sur les collections dans la base de données `test`.

Pour voir la base de données dans laquelle vous vous trouvez ou, en d'autres termes, pour émettre des commandes, utilisez la db commande dans le shell mongo, comme suit.

Interrogation :

```
db
```

Sortie :

```
test
```

Bien que la connexion par défaut puisse être associée au contexte de la base de données `test`, cela ne signifie pas nécessairement que l'utilisateur associé à la connexion est autorisé à effectuer des actions sur la base de données `test`. Dans l'exemple de scénario précédent, si vous vous authentifiez en tant qu'utilisateur `user3`, auquel le rôle `readWrite` a été attribué pour la base de données `sample-database-1`, le contexte par défaut de votre connexion est la base de données `test`. Toutefois, si vous essayez d'insérer un document dans une collection de la base de données `test`, un message d'erreur d'échec d'autorisation s'affiche. En effet, cet utilisateur n'est pas autorisé à exécuter cette commande sur cette base de données, comme indiqué ci-dessous.

Interrogation :

```
db
```

Sortie :

```
test
```

Interrogation :

```
db.col.insert({x:1})
```

Sortie :

```
WriteCommandError({ "ok" : 0, "code" : 13, "errmsg" : "Authorization failure" })
```

Si vous modifiez le contexte de votre connexion à la base de données `sample-database-1`, vous pouvez écrire dans la collection pour laquelle l'utilisateur a reçu l'autorisation.

Interrogation :

```
use sample-database-1
```

Sortie :

```
switched to db sample-database-1
```

Interrogation :

```
db.col.insert({x:1})
```

Sortie :

```
WriteResult({ "nInserted" : 1})
```

Lorsque vous vous authentifiez auprès d'un cluster avec un utilisateur particulier, vous pouvez également spécifier la base de données dans la chaîne de connexion. Cela supprime la nécessité d'exécuter la commande `use` après l'authentification de l'utilisateur sur la base de données `admin`.

La chaîne de connexion suivante authentifie l'utilisateur par rapport à la base de données `admin`, mais le contexte de la connexion sera celui de la base de données `sample-database-1`.

```
mongo "mongodb://user3:abc123@sample-cluster.node.us-east-1.docdb.amazonaws.com:27017/
sample-database-2"
```

Commandes courantes

Cette section fournit des exemples de commandes courantes utilisant le contrôle d'accès basé sur les rôles dans Amazon DocumentDB. Vous devez être dans le contexte de la base de données `admin` pour créer et modifier des utilisateurs et des rôles. Vous pouvez utiliser la commande `use admin` pour basculer vers la base de données `admin`.

> **Note**
>
> Les modifications apportées aux utilisateurs et aux rôles se produiront implicitement dans la base de données `admin`. La création d'utilisateurs avec des rôles étendus à toutes les bases de données (par exemple, `readAnyDatabase`) nécessite que vous soyez soit dans le contexte de la base de données `admin` (c'est-à-dire, `use admin`) lors de la création de l'utilisateur, ou que vous indiquiez explicitement la base de données pour le rôle lors de la création de l'utilisateur (comme indiqué dans l'exemple 2 de cette section).

Exemple 1 : créer un utilisateur avec un `read` rôle pour la base de données`foo`.

```
db.createUser({user: "readInFooBar", pwd: "abc123", roles: [{role: "read", db:
 "foo"}]})
```

Le résultat de cette opération ressemble à ceci.

```
{
   "user":"readInFooBar",
   "roles":[
      {
         "role":"read",
         "db":"foo"
      }
   ]
}
```

Exemple 2 : créez un utilisateur avec un accès en lecture sur toutes les bases de données.

```
db.createUser({user: "readAllDBs", pwd: "abc123", roles: [{role: "readAnyDatabase", db:
 "admin"}]})
```

Le résultat de cette opération ressemble à ceci.

```
{
   "user":"readAllDBs",
   "roles":[
      {
         "role":"readAnyDatabase",
         "db":"admin"
      }
   ]
}
```

Exemple 3 : Accorder un `read` rôle à un utilisateur existant sur une nouvelle base de données.

```
db.grantRolesToUser("readInFooBar", [{role: "read", db: "bar"}])
```

Exemple 4 : mettre à jour le rôle d'un utilisateur

```
db.updateUser("readInFooBar", {roles: [{role: "read", db: "foo"}, {role: "read", db:
 "baz"}]})
```

Exemple 5 : Révoquer l'accès à une base de données pour un utilisateur.

```
db.revokeRolesFromUser("readInFooBar", [{role: "read", db: "baz"}])
```

Exemple 6 : Décrivez un rôle intégré.

```
db.getRole("read", {showPrivileges:true})
```

Le résultat de cette opération ressemble à ceci.

```
{
   "role":"read",
   "db":"sample-database-1",
   "isBuiltin":true,
   "roles":[

   ],
   "inheritedRoles":[

   ],
   "privileges":[
      {
         "resource":{
            "db":"sample-database-1",
            "collection":""
         },
         "actions":[
            "changeStream",
            "collStats",
            "dbStats",
            "find",
            "killCursors",
            "listCollections",
            "listIndexes"
         ]
      }
   ],
   "inheritedPrivileges":[
      {
```

```
         "resource":{
            "db":"sample-database-1",
            "collection":""
         },
         "actions":[
            "changeStream",
            "collStats",
            "dbStats",
            "find",
            "killCursors",
            "listCollections",
            "listIndexes"
         ]
      }
   }
```

Exemple 7 : Supprimer un utilisateur du cluster.

```
db.dropUser("readInFooBar")
```

Le résultat de cette opération ressemble à ceci.

```
true
```

Exemple 8 : Création d'un rôle avec accès en lecture et en écriture à une collection spécifique

```
db.createRole(
{
    role: "collectionRole",
    privileges: [
    {
        resource: {db: "sample-database-2", collection: "col2"}, actions: ["find",
 "update", "insert", "remove"]
    }],
    roles: []
}
)
```

Le résultat de cette opération ressemble à ceci.

```
{
   "role":"collectionRole",
```

```
   "privileges":[
      {
         "resource":{
            "db":"sample-database-2",
            "collection":"col2"
         },
         "actions":[
            "find",
            "update",
            "insert",
            "remove"
         ]
      }
   ],
   "roles":[

   ]
}
```

Exemple 9 : Création d'un utilisateur et attribution d'un rôle défini par l'utilisateur

```
db.createUser(
{
    user: "user3",
    pwd: "abc123",
    roles: []
})

db.grantRolesToUser("user3",["collectionRole"])
```

Exemple 10 : Accorder des privilèges supplémentaires à un rôle défini par l'utilisateur

```
db.grantPrivilegesToRole(
  "collectionRole",
  [
    {
      resource: { db: "sample-database-1", collection: "col1" },
      actions: ["find", "update", "insert", "remove"]
    }
  ]
)
```

Exemple 11 : Supprimer les privilèges d'un rôle défini par l'utilisateur

```
db.revokePrivilegesFromRole(
  "collectionRole",
  [
    {
      resource: { db: "sample-database-1", collection: "col2" },
      actions: ["find", "update", "insert", "remove"]
    }
  ]
)
```

Exemple 12 : Mettre à jour un rôle défini par l'utilisateur existant

```
db.updateRole(
  "collectionRole",
  {
    privileges: [
    {
        resource: {db: "sample-database-3", collection: "sample-collection-3"},
 actions: ["find", "update", "insert", "remove"]
    }],
    roles: []
  }
)
```

Différences fonctionnelles

Dans Amazon DocumentDB, les définitions des utilisateurs et des rôles sont stockées dans la `admin` base de données et les utilisateurs sont authentifiés par rapport à la base de données. `admin` Cette fonctionnalité diffère de MongoDB Community Edition, mais est compatible avec MongoDB Atlas.

Amazon DocumentDB prend également en charge les flux de modifications, qui fournissent une séquence chronologique des événements de modification qui se produisent dans les collections de votre cluster. L'`listChangeStreams`action est appliquée au niveau du cluster (c'est-à-dire dans toutes les bases de données), et l'`modifyChangeStreams`action peut être appliquée au niveau de la base de données et au niveau du cluster.

Limites

Le tableau suivant contient les limites du contrôle d'accès basé sur les rôles dans Amazon DocumentDB.

Description	Limite
Nombre d'utilisateurs par cluster	1 000
Nombre de rôles associés à un utilisateur	1 000
Nombre de rôles définis par l'utilisateur	100
Nombre de ressources associées à un privilège	100

Accès à la base de données à l'aide du contrôle d'accès basé sur les rôles

Avec le contrôle d'accès basé sur les rôles, vous pouvez créer un utilisateur et lui accorder un ou plusieurs rôles afin de déterminer les opérations qu'il peut effectuer dans une base de données ou un cluster.

Voici une liste des rôles intégrés actuellement pris en charge dans Amazon DocumentDB.

> **Note**
>
> Dans Amazon DocumentDB 4.0 et 5.0, les `ListDatabase` commandes `ListCollection` and peuvent éventuellement utiliser les `authorizedDatabases` paramètres `authorizedCollections` and pour répertorier les collections et les bases de données auxquelles l'utilisateur est autorisé à accéder en exigeant les `listDatabase` rôles `listCollections` and, respectivement. De plus, les utilisateurs ont désormais la possibilité de supprimer leurs propres curseurs sans avoir besoin du `KillCursor` rôle.

Database user

Nom du rôle	Description	Actions

`read`	Accorde à un utilisateur un accès en lecture à la base de données spécifiée.	`changeStreams`
		`collStats`
		`dbStats`
		`find`

Nom du rôle	Description	Actions

		`killCursors`
		
		`listIndexes`
		
		`listCollections`

`readWrite`	Accorde à l'utilisateur un accès en lecture et en écriture à la base de données spécifiée.	Toutes les actions à partir des autorisations `read`.
		
		`createCollection`
		
		`dropCollection`
		
		`createIndex`
		
		`dropIndex`
		
		`insert`
		
		`killCursors`
		
		`listIndexes`
		
		`listCollections`
		
		`remove`
		
		`update`

Cluster user

Nom du rôle	Description	Actions

readAnyDatabase	Accorde à un utilisateur un accès en lecture à toutes les bases de données du cluster.	Toutes les actions à partir des autorisations `read`.
		`listChangeStreams`
		`listDatabases`

`readWriteAnyDatabase`	Accorde à un utilisateur un accès en lecture et en écriture à toutes les bases de données du cluster.	Toutes les actions à partir des autorisations `readWrite` .
		`listChangeStreams`
		`listDatabases`

`userAdminAnyDatabase`	Accorde à un utilisateur la possibilité d'attribuer et de modifier les rôles ou les privilèges dont un utilisateur dispose pour la base de données spécifiée.	`changeCustomData`
		`changePassword`
		`createUser`
		`dropRole`
		`dropUser`
		`grantRole`
		`listDatabases`
		`revokeRole`
		`viewRole`

Nom du rôle	Description	Actions
		`viewUser`

`dbAdminAnyDatabase`	Accorde à un utilisateur la possibilité d'exécuter des rôles d'administration de base de données sur une base de données spécifiée.	Toutes les actions à partir des autorisations `dbAdmin`.
		`dropCollection`
		`listDatabases`
		`listChangeStreams`
		`modifyChangeStreams`

Superuser

Nom du rôle	Description	Actions
`root`	Accorde à un utilisateur l'accès aux ressources et aux opérations de tous les rôles suivants combinés : `readWriteAnyDatabase`, `dbAdminAnyDatabase`, `userAdminAnyDatabase`, `clusterAdmin`, `restore` et `backup`.	Toutes les actions à partir de `readWriteAnyDatabase`, `dbAdminAnyDatabase`, `userAdminAnyDatabase`, `clusterAdmin`, `restore` et `backup`.

Database administrator

Nom du rôle	Description	Actions
dbAdmin	Accorde à un utilisateur la possibilité d'effectuer des tâches d’administration sur la base de données spécifiée.	bypassDocumentValidation collMod collStats createCollection createIndex dropCollection dropDatabase dropIndex dbStats find killCursors listIndexes listCollections modifyChangeStreams
dbOwner	Accorde à un utilisateur la possibilité d'effectuer des tâches d’administration sur la base de données spécifiée en combinant les rôles dbAdmin et readWrite .	Toutes les actions à partir de dbAdmin et readWrite .

Cluster administrator

Nom du rôle	Description	Actions

`clusterAdmin`	Accorde à un utilisateur l'accès de gestion de cluster le plus élargi en combinant les rôles `hostManager` , `clusterManager` et `clusterMonitor` .	Toutes les actions à partir de `clusterManager` , `clusterMonitor` et `hostManager` .
		`listChangeStreams`
		`dropDatabase`
		`modifyChangeStreams`

`clusterManager`	Accorde à un utilisateur la possibilité d'effectuer des actions de gestion et de surveillance sur le cluster spécifié.	`listChangeStreams`
		`listSessions`
		`modifyChangeStreams`
		`replSetGetConfig`

`clusterMonitor`	Accorde à un utilisateur la possibilité d'accéder en lecture seule aux outils de surveillance.	`collStats`
		`dbStats`
		`find`
		`getParameter`
		`hostInfo`
		`indexStats`

Nom du rôle	Description	Actions
		killCursors listChangeStreams listCollections listDatabases listIndexes listSessions replSetGetConfig serverStatus top
hostManager	Accorde à un utilisateur la possibilité de surveiller et de gérer les serveurs.	killCursors killAnyCursor killAnySession killop

Backup administrator

Nom du rôle	Description	Actions
backup	Accorde à un utilisateur l'accès nécessaire à la sauvegarde des données.	getParameter insert find

Nom du rôle	Description	Actions

		`listChangeStreams`
		`listCollections`
		`listDatabases`
		`listIndexes`
		`update`

Nom du rôle	Description	Actions

`restore`	Accorde à un utilisateur l'accès nécessaire à la restauration des données.	`bypassDocumentValidation`
		`changeCustomData`
		`changePassword`
		`collMod`
		`createCollection`
		`createIndex`
		`createUser`
		`dropCollection`
		`dropRole`
		`dropUser`
		`getParameter`
		`grantRole`
		`find`
		`insert`
		`listCollections`
		`modifyChangeStreams`
		`revokeRole`

Nom du rôle	Description	Actions

		`remove`
		
		`viewRole`
		
		`viewUser`
		
		`update`

Journalisation et surveillance dans Amazon DocumentDB

Amazon DocumentDB (avec compatibilité avec MongoDB) propose diverses CloudWatch métriques Amazon que vous pouvez surveiller pour déterminer l'intégrité et les performances de vos clusters et instances Amazon DocumentDB. Vous pouvez consulter les métriques Amazon DocumentDB à l'aide de divers outils, notamment la console Amazon DocumentDBAWS CLI, la CloudWatch console Amazon et l' CloudWatch API. Pour de plus amples informations sur la surveillance, veuillez consulter Surveillance Amazon DocumentDB.

Outre les CloudWatch métriques Amazon, vous pouvez utiliser le profileur pour enregistrer le temps d'exécution et les détails des opérations effectuées sur votre cluster. Le profileur est utile pour surveiller les opérations les plus lentes sur votre cluster afin de vous aider à améliorer les performances des requêtes individuelles et les performances globales du cluster. Lorsque cette option est activée, les opérations sont enregistrées dans Amazon CloudWatch Logs et vous pouvez utiliser CloudWatch Insight pour analyser, surveiller et archiver vos données de profilage Amazon DocumentDB. Pour plus d'informations, veuillez consulter Profilage des opérations Amazon DocumentDB.

Amazon DocumentDB est également intégré avecAWS CloudTrail, service qui enregistre les actions effectuées par les utilisateurs, les rôles ou unAWS service dans Amazon DocumentDB (avec compatibilité avec MongoDB). CloudTrail capture tous les appels d'AWS CLIAPI pour Amazon DocumentDB en tant qu'événements, y compris les appels Amazon DocumentDBAWS Management Console et les appels de code à Amazon DocumentDB. Pour plus d'informations, veuillez consulter Journalisation des appels d'API Amazon DocumentDB à l'aide d'AWS CloudTrail.

Avec Amazon DocumentDB, vous pouvez auditer les événements qui ont été réalisés dans votre cluster. Les exemples d'événements enregistrés incluent les tentatives d'authentification réussies et celles ayant échoué, la suppression d'une collection dans une base de données ou la création

d'un index. Par défaut, l'audit est désactivé sur Amazon DocumentDB et vous devez activer cette fonctionnalité. Pour plus d'informations, veuillez consulter Audit des événements Amazon DocumentDB.

Mise à jour de vos certificats TLS Amazon DocumentDB

Rubriques

Le certificat de l'autorité de certification (CA) pour les clusters Amazon DocumentDB sera mis à jour à partir d'août 2024. Si vous utilisez des clusters Amazon DocumentDB avec le protocole TLS (Transport Layer Security) activé (paramètre par défaut) et que vous n'avez pas alterné vos certificats d'application client et de serveur, les étapes suivantes sont nécessaires pour atténuer les problèmes de connectivité entre votre application et vos clusters Amazon DocumentDB.

Les certificats de l'autorité de certification et du serveur ont été mis à jour dans le cadre des meilleures pratiques de maintenance et de sécurité standard pour Amazon DocumentDB. Les applications clientes doivent ajouter les nouveaux certificats CA à leurs magasins de confiance, et les instances Amazon DocumentDB existantes doivent être mises à jour pour utiliser les nouveaux certificats CA avant cette date d'expiration.

Mise à jour de votre application et de votre cluster Amazon DocumentDB

Suivez les étapes de cette section pour mettre à jour votre ensemble de certificats d'autorité de certification de votre application (Étape 1) et les certificats de serveur de votre cluster (Étape 2). Avant d'appliquer les modifications à vos environnements de production, nous vous recommandons fortement de tester ces étapes dans un environnement de développement ou de transit.

> **Note**
>
> Vous devez effectuer les étapes 1 et 2 dans chacune des étapes Région AWS dans lesquelles vous avez des clusters Amazon DocumentDB.

Étape 1 : Télécharger le nouveau certificat d'autorité de certification et mettre à jour votre application

Téléchargez le nouveau certificat CA et mettez à jour votre application pour utiliser le nouveau certificat CA afin de créer des connexions TLS avec Amazon DocumentDB. Téléchargez le nouveau lot de certificats CA à partir de https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem. Cette opération entraîne le téléchargement d'un fichier nommé `global-bundle.pem`.

> **Note**
>
> Si vous accédez au keystore qui contient à la fois l'ancien certificat CA (`rds-ca-2019-root.pem`) et les nouveaux certificats CA (`rds-ca-rsa2048-g1, rds-ca-rsa4096-g1`), vérifiez que le keystore est sélectionné. `global-bundle`

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

Ensuite, mettez à jour vos applications pour utiliser le nouveau lot de certificats. Le nouveau bundle CA contient à la fois l'ancien certificat CA (rds-ca-2019) et les nouveaux certificats CA (rds-ca-rsa2048-g1, 4096-g1). rds-ca-rsa Le fait d'avoir les deux certificats d'autorité de certification dans le nouveau lot de l'autorité de certification vous permet de mettre à jour votre application et votre cluster en deux étapes.

Pour vérifier que votre application utilise le dernier lot de certificats de l'autorité de certification, veuillez consulter Comment puis-je être sûr d'utiliser le tout dernier pack CA ?. Si vous utilisez déjà le dernier lot de certificats de l'autorité de certification dans votre application, vous pouvez passer à l'étape 2.

Pour obtenir des exemples d'utilisation d'une offre groupée CA avec votre application, consultez Chiffrement des données en transit et Connexion avec TLS activé.

> **Note**
>
> Actuellement, le pilote MongoDB Go 1.2.1 accepte uniquement un certificat de serveur d'autorité de certification dans `sslcertificateauthorityfile`. Veuillez consulter Connexion avec TLS activé pour vous connecter à Amazon DocumentDB à l'aide de Go lorsque TLS est activé.

Étape 2 : Mettre à jour le certificat de serveur

Une fois que l'application a été mise à jour pour utiliser le nouveau bundle CA, l'étape suivante consiste à mettre à jour le certificat du serveur en modifiant chaque instance d'un cluster Amazon DocumentDB. Pour modifier les instances afin qu'elles utilisent le nouveau certificat de serveur, consultez les instructions suivantes.

Amazon DocumentDB fournit les autorités de certification suivantes pour signer le certificat de serveur de base de données pour une instance de base de données :

- rds-ca-rsa2048-g1 —Utilise une autorité de certification avec l'algorithme de clé privée RSA 2048 et l'algorithme de signature SHA256 dans la plupart des régions. AWS Cette autorité de certification prend en charge la rotation automatique des certificats de serveur.
- rds-ca-rsa4096-g1 —Utilise une autorité de certification avec l'algorithme de clé privée RSA 4096 et l'algorithme de signature SHA384. Cette autorité de certification prend en charge la rotation automatique des certificats de serveur.

> **Note**
>
> Si vous utilisez le AWS CLI, vous pouvez vérifier la validité des autorités de certification répertoriées ci-dessus en utilisant describe-certificates.

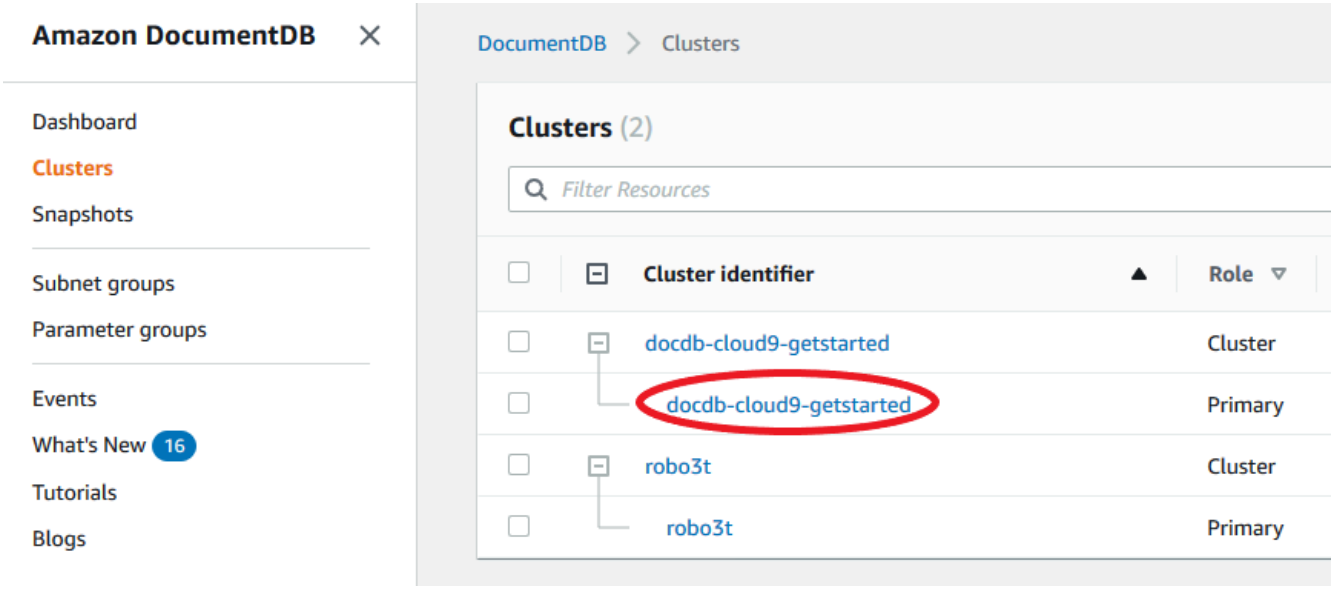
Ces certificats de CA sont inclus dans la solution groupée de certificats régionaux et mondiaux. Lorsque vous utilisez l'autorité de certification rds-ca-rsa 2048-g1 ou rds-ca-rsa 4096-g1 avec une base de données, Amazon DocumentDB gère le certificat du serveur de base de données sur la base de données. Amazon DocumentDB fait automatiquement pivoter le certificat du serveur de base de données avant son expiration (un redémarrage peut être nécessaire).

> **Note**
>
> La mise à jour de vos instances nécessite un redémarrage, ce qui peut entraîner une interruption du service. Vous devez avoir terminé l'Étape 1 avant de mettre à jour le certificat de serveur.

Using the AWS Management Console

Procédez comme suit pour identifier et faire pivoter l'ancien certificat de serveur pour vos instances Amazon DocumentDB existantes à l'aide du. AWS Management Console

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans la liste des régions située dans le coin supérieur droit de l'écran, choisissez celle Région AWS dans laquelle résident vos clusters.
3. Dans le volet de navigation sur le côté gauche de la console, choisissez Clusters.
4. Vous devrez peut-être identifier les instances qui figurent toujours sur l'ancien certificat de serveur (`rds-ca-2019`). Vous pouvez le faire dans la colonne Autorité de certification située à l'extrême droite du tableau Clusters.
5. Dans le tableau des clusters, vous verrez la colonne Identifiant du cluster à l'extrême gauche. Vos instances sont répertoriées sous des clusters, comme dans la capture d'écran ci-dessous.



6. Cochez la case située à gauche de l'instance qui vous intéresse.

7. Choisissez Actions, puis Modifier.
8. Sous Autorité de certification, sélectionnez le nouveau certificat de serveur (c.-à-d., `rds-ca-rsa2048-g1`) pour cette instance.
9. Vous pouvez voir un résumé des modifications à la page suivante. Notez qu'il y a une alerte supplémentaire pour vous rappeler de vous assurer que votre application utilise le dernier pack CA de certificats avant de modifier l'instance afin d'éviter d'interrompre la connectivité.
10. Vous pouvez choisir d'appliquer la modification lors de votre prochaine fenêtre de maintenance ou de l'appliquer immédiatement. Si vous avez l'intention de modifier immédiatement le certificat de serveur, utilisez l'option Apply Immediately (Appliquer immédiatement).
11. Choisissez Modify instance (Modifier l'instance) pour terminer la mise à jour.

Using the AWS CLI

Procédez comme suit pour identifier et faire pivoter l'ancien certificat de serveur pour vos instances Amazon DocumentDB existantes à l'aide du. AWS CLI

1. Pour modifier immédiatement les instances, exécutez la commande suivante pour chaque instance du cluster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa2048-g1 --apply-immediately
```

2. Pour modifier les instances de vos clusters afin d'utiliser le nouveau certificat d'autorité de certification lors de la prochaine fenêtre de maintenance de votre cluster, exécutez la commande suivante pour chaque instance du cluster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa2048-g1 --no-apply-immediately
```

Résolution des problèmes

Si vous rencontrez des problèmes de connexion à votre cluster dans le cadre de la rotation du certificat, nous vous suggérons de procéder comme suit :

- Redémarrez vos instances. La rotation du nouveau certificat nécessite le redémarrage de chacune de vos instances. Si vous avez appliqué le nouveau certificat à une ou plusieurs instances mais

que vous ne les avez pas redémarrées, redémarrez vos instances pour appliquer le nouveau certificat. Pour plus d'informations, consultez Redémarrage d'une instance Amazon DocumentDB.

- Vérifiez que vos clients utilisent le dernier ensemble de certificats. veuillez consulter Comment puis-je être sûr d'utiliser le tout dernier pack CA ?.
- Vérifiez que vos instances utilisent le dernier certificat. veuillez consulter Comment savoir laquelle de mes instances Amazon DocumentDB utilise l'ancien ou le nouveau certificat de serveur ?.
- Vérifiez que la dernière autorité de certification est utilisée par votre application. Certains pilotes, comme Java et Go, nécessitent un code supplémentaire pour importer plusieurs certificats à partir d'un ensemble de certificats vers le magasin de confiance. Pour plus d'informations sur la connexion à Amazon DocumentDB via TLS, consultez. Connexion par programmation à Amazon DocumentDB
- Contactez le support. Si vous avez des questions ou des problèmes, contactez AWS Support.

Questions fréquentes (FAQ)

Voici les réponses à certaines questions courantes concernant les certificats TLS.

Que faire si j'ai des questions ou des problèmes ?

Si vous avez des questions ou des problèmes, contactez AWS Support.

Comment savoir si j'utilise le protocole TLS pour me connecter à mon cluster Amazon DocumentDB ?

Vous pouvez déterminer si votre cluster utilise TLS en examinant le paramètre `tls` du groupe de paramètres de cluster de votre cluster. Si le paramètre `tls` est défini sur `enabled`, vous utilisez le certificat TLS pour vous connecter à votre cluster. Pour plus d'informations, consultez Gestion des groupes de paramètres du cluster Amazon DocumentDB.

Pourquoi mettez-vous à jour les certificats d'autorité de certification et de serveur ?

L'autorité de certification Amazon DocumentDB et les certificats de serveur sont mis à jour dans le cadre des meilleures pratiques de maintenance et de sécurité standard pour Amazon DocumentDB. Les certificats de CA et de serveur actuels expirent à compter du mois d'août 2024.

Que se passe-t-il si je ne prends aucune mesure avant la date d'expiration ?

Si vous utilisez le protocole TLS pour vous connecter à votre cluster Amazon DocumentDB et que vous ne modifiez pas le certificat avant le vos applications connectées via TLS ne pourront plus communiquer avec le cluster Amazon DocumentDB.

Amazon DocumentDB ne fera pas automatiquement pivoter vos certificats de base de données avant leur expiration. Vous devez mettre à jour vos applications et vos clusters pour utiliser les nouveaux certificats CA avant ou après la date d'expiration.

Comment savoir laquelle de mes instances Amazon DocumentDB utilise l'ancien ou le nouveau certificat de serveur ?

Pour identifier les instances Amazon DocumentDB qui utilisent toujours l'ancien certificat de serveur, vous pouvez utiliser Amazon AWS Management Console DocumentDB ou le. AWS CLI

À l'aide du AWS Management Console

Pour identifier les instances de vos clusters qui utilisent l'ancien certificat

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans la liste des régions située dans le coin supérieur droit de l'écran, choisissez celle Région AWS dans laquelle résident vos instances.
3. Dans le volet de navigation sur le côté gauche de la console, choisissez Clusters.
4. La colonne Autorité de certification (à l'extrême droite du tableau) indique les instances qui figurent toujours sur l'ancien certificat de serveur (`rds-ca-2019`) et sur le nouveau certificat de serveur (`rds-ca-rsa2048-g1`).

À l'aide du AWS CLI

Pour identifier les instances de vos clusters qui utilisent l'ancien certificat de serveur, utilisez la commande `describe-db-clusters` avec les éléments suivants.

```
aws docdb describe-db-instances \
    --filters Name=engine,Values=docdb \
    --query 'DBInstances[*].
{CertificateVersion:CACertificateIdentifier,InstanceID:DBInstanceIdentifier}'
```

Comment modifier les instances individuelles de mon cluster Amazon DocumentDB pour mettre à jour le certificat du serveur ?

Nous vous recommandons de mettre à jour simultanément les certificats de serveur pour toutes les instances d'un cluster donné. Pour modifier les instances de votre cluster, vous pouvez utiliser la console ou l' AWS CLI.

> **Note**
>
> La mise à jour de vos instances nécessite un redémarrage, ce qui peut entraîner une interruption du service. Vous devez avoir terminé l'Étape 1 avant de mettre à jour le certificat de serveur.

À l'aide du AWS Management Console

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans la liste des régions située dans le coin supérieur droit de l'écran, choisissez celle Région AWS dans laquelle résident vos clusters.
3. Dans le volet de navigation sur le côté gauche de la console, choisissez Clusters.
4. La colonne Autorité de certification (à l'extrême droite du tableau) indique les instances qui figurent toujours sur l'ancien certificat de serveur (`rds-ca-2019`).
5. Dans le tableau Clusters, sous Identifiant du cluster, sélectionnez une instance à modifier.
6. Choisissez Actions, puis Modifier.
7. Sous Autorité de certification, sélectionnez le nouveau certificat de serveur (c.-à-d., `rds-ca-rsa2048-g1`) pour cette instance.
8. Vous pouvez voir un résumé des modifications à la page suivante. Notez qu'il y a une alerte supplémentaire pour vous rappeler de vous assurer que votre application utilise le dernier pack CA de certificats avant de modifier l'instance afin d'éviter d'interrompre la connectivité.
9. Vous pouvez choisir d'appliquer la modification lors de votre prochaine fenêtre de maintenance ou de l'appliquer immédiatement.
10. Choisissez Modify instance (Modifier l'instance) pour terminer la mise à jour.

À l'aide du AWS CLI

Procédez comme suit pour identifier et faire pivoter l'ancien certificat de serveur pour vos instances Amazon DocumentDB existantes à l'aide du. AWS CLI

1. Pour modifier immédiatement les instances, exécutez la commande suivante pour chaque instance du cluster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa2048-g1 --apply-immediately
```

2. Pour modifier les instances de vos clusters afin d'utiliser le nouveau certificat d'autorité de certification lors de la prochaine fenêtre de maintenance de votre cluster, exécutez la commande suivante pour chaque instance du cluster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa2048-g1 --no-apply-immediately
```

Que se passe-t-il si j'ajoute une nouvelle instance à un cluster existant ?

Toutes les nouvelles instances créées utilisent l'ancien certificat de serveur et nécessitent des connexions TLS à l'aide de l'ancien certificat d'autorité de certification. Toutes les nouvelles instances Amazon DocumentDB créées après le 25 janvier 2024 utiliseront par défaut le nouveau certificat rds-ca-rsa 2048-g1.

Que se passe-t-il s'il y a un remplacement d'instance ou un basculement sur incident sur mon cluster ?

S'il y a un remplacement d'instance dans votre cluster, la nouvelle instance créée continue d'utiliser le même certificat de serveur que celui que l'instance utilisait précédemment. Nous vous recommandons de mettre à jour les certificats de serveur pour toutes les instances en même temps. Si un basculement se produit dans le cluster, le certificat de serveur sur le nouveau serveur principal est utilisé.

Si je n'utilise pas TLS pour me connecter à mon cluster, dois-je toujours mettre à jour chacune de mes instances ?

Si vous n'utilisez pas le protocole TLS pour vous connecter à vos clusters Amazon DocumentDB, aucune action n'est nécessaire.

Si je n'utilise pas TLS pour me connecter à mon cluster mais que je prévois de le faire à l'avenir, que dois-je faire ?

Si vous avez créé un cluster avant janvier 2024, suivez les étapes 1 et 2 de la section précédente pour vous assurer que votre application utilise le bundle CA mis à jour et que chaque instance Amazon DocumentDB utilise le dernier certificat de serveur. Si vous créez un cluster après le 25 janvier 2024, celui-ci disposera déjà du dernier certificat de serveur (rds-ca-rsa2048-g1). Pour vérifier que votre application utilise le dernier ensemble de certificats d'autorité de certification, consultez Si je n'utilise pas TLS pour me connecter à mon cluster, dois-je toujours mettre à jour chacune de mes instances ?.

La date limite peut-elle être prolongée au-delà d'août 2024 ?

Si vos applications se connectent via TLS, le délai ne peut pas être prolongé.

Comment puis-je être sûr d'utiliser le tout dernier pack CA ?

Pour vérifier que vous disposez du bundle le plus récent, utilisez la commande suivante. Pour exécuter cette commande, Java doit être installé et les outils Java doivent se trouver dans la variable PATH de votre shell. Pour plus d'informations, voir Utilisation de Java

macOS et Amazon Linux

```
keytool -printcert -v -file global-bundle.pem
```

Windows

```
keytool -printcert -v -file global-bundle.p7b
```

Pourquoi est-ce que je vois « RDS » dans le nom du groupe CA ?

Pour certaines fonctionnalités de gestion, telles que la gestion des certificats, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon Relational Database Service (Amazon RDS).

Quand le nouveau certificat expirera-t-il ?

Le nouveau certificat de serveur expirera (généralement) comme suit :

- rds-ca-rsa2048-g1 — Expire en 2016

- rds-ca-rsa4096-g1 — Expire en 2121

Si j'ai appliqué le nouveau certificat de serveur, puis-je revenir à l'ancien certificat ?

Si vous devez rétablir une instance à l'ancien certificat de serveur, nous vous recommandons de le faire pour toutes les instances du cluster. Vous pouvez rétablir le certificat de serveur pour chaque instance d'un cluster en utilisant le AWS Management Console ou le AWS CLI.

À l'aide du AWS Management Console

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans la liste des régions située dans le coin supérieur droit de l'écran, choisissez celle Région AWS dans laquelle résident vos clusters.
3. Dans le volet de navigation sur le côté gauche de la console, choisissez Clusters.
4. Dans le tableau Clusters, sous Identifiant du cluster, sélectionnez une instance à modifier. Choisissez Actions, puis Modify (Modifier).
5. Sous Autorité de certification, vous pouvez sélectionner l'ancien certificat de serveur (`rds-ca-2019`).
6. Sélectionnez Continuer pour afficher un résumé de vos modifications.
7. Dans cette page, vous pouvez choisir de planifier l'application de vos modifications dans la prochaine fenêtre de maintenance ou d'appliquer vos modifications immédiatement. Effectuez votre sélection et choisissez Modify instance (Modifier l'instance).

> **Note**
>
> Si vous choisissez d'appliquer les modifications immédiatement, les modifications placées dans la file d'attente des modifications en attente sont également appliquées. Si des modifications en attente ont besoin d'un temps d'arrêt, choisir de les appliquer immédiatement peut entraîner un temps d'arrêt imprévu.

À l'aide du AWS CLI

```
aws docdb modify-db-instance --db-instance-identifier <db_instance_name> ca-
certificate-identifier rds-ca-2019 <--apply-immediately | --no-apply-immediately>
```

Si vous choisissez `--no-apply-immediately`, les modifications seront appliquées lors de la prochaine fenêtre de maintenance du cluster.

Si nous effectuons la restauration à partir d'un instantané ou d'un instant dans le passé, aura-t-il le nouveau certificat de serveur ?

Si vous restaurez un instantané ou si vous effectuez une point-in-time restauration après août 2024, le nouveau cluster créé utilisera le nouveau certificat CA.

Que faire si je rencontre des problèmes pour me connecter directement à mon cluster Amazon DocumentDB depuis n'importe quel système d'exploitation Mac OS ?

Mac OS a mis à jour les exigences relatives aux certificats sécurisés. Les certificats fiables doivent désormais être valides pendant 397 jours ou moins (voirhttps://support.apple.com/en-us/HT211025).

> **Note**
>
> Cette restriction est observée dans les nouvelles versions de Mac OS.

Les certificats d'instance Amazon DocumentDB sont valides pendant plus de quatre ans, soit plus que la durée maximale autorisée pour Mac OS. Pour vous connecter directement à un cluster Amazon DocumentDB depuis un ordinateur exécutant Mac OS, vous devez autoriser les certificats non valides lors de la création de la connexion TLS. Dans ce cas, les certificats non valides signifient que la période de validité est supérieure à 397 jours. Vous devez comprendre les risques avant d'autoriser des certificats non valides lors de la connexion à votre cluster Amazon DocumentDB.

Pour vous connecter à un cluster Amazon DocumentDB depuis Mac OS à l'aide du paramètre AWS CLI, utilisez le `tlsAllowInvalidCertificates` paramètre.

```
mongo --tls --host <hostname> --username <username> --password <password> --port 27017
 --tlsAllowInvalidCertificates
```

Mise à jour de vos certificats TLS Amazon DocumentDB — (USA Ouest) GovCloud

> **Note**
>
> Ces informations s'appliquent uniquement aux utilisateurs de la région GovCloud (ouest des États-Unis).

Le certificat d'autorité de certification (CA) pour les clusters Amazon DocumentDB (compatible avec MongoDB) sera mis à jour le 18 mai 2022. Si vous utilisez des clusters Amazon DocumentDB avec le protocole TLS (Transport Layer Security) activé (paramètre par défaut) et que vous n'avez pas alterné vos certificats d'application client et de serveur, les étapes suivantes sont nécessaires pour atténuer les problèmes de connectivité entre votre application et vos clusters Amazon DocumentDB.

- Étape 1 : Télécharger le nouveau certificat d'autorité de certification et mettre à jour votre application
- Étape 2 : Mettre à jour le certificat de serveur

Les certificats de l'autorité de certification et du serveur ont été mis à jour dans le cadre des meilleures pratiques de maintenance et de sécurité standard pour Amazon DocumentDB. Le certificat CA précédent expirera le 18 mai 2022. Les applications clientes doivent ajouter les nouveaux certificats CA à leurs magasins de confiance, et les instances Amazon DocumentDB existantes doivent être mises à jour pour utiliser les nouveaux certificats CA avant cette date d'expiration.

Mise à jour de votre application et de votre cluster Amazon DocumentDB

Suivez les étapes de cette section pour mettre à jour votre ensemble de certificats d'autorité de certification de votre application (Étape 1) et les certificats de serveur de votre cluster (Étape 2). Avant d'appliquer les modifications à vos environnements de production, nous vous recommandons fortement de tester ces étapes dans un environnement de développement ou de transit.

> **Note**
>
> Vous devez effectuer les étapes 1 et 2 pour chacune des étapes Région AWS dans lesquelles vous avez des clusters Amazon DocumentDB.

Étape 1 : Télécharger le nouveau certificat d'autorité de certification et mettre à jour votre application

Téléchargez le nouveau certificat CA et mettez à jour votre application pour utiliser le nouveau certificat CA afin de créer des connexions TLS avec Amazon DocumentDB. Téléchargez le nouveau lot de certificats CA à partir de https://truststore.pki.us-gov-west-1.rds.amazonaws.com/us-gov-west-1/us-gov-west-1-bundle.pem. Cette opération entraîne le téléchargement d'un fichier nommé `us-gov-west-1-bundle.pem`.

> **Note**
>
> Si vous accédez au keystore (stockage de clés) qui contient à la fois l'ancien certificat de l'autorité de certification (`rds-ca-2017-root.pem`) et le nouveau certificat de l'autorité de certification (`rds-ca-rsa4096-g1.pem`), vérifiez que le keystore sélectionne `CA-RSA4096-G1`.

```
wget https://truststore.pki.us-gov-west-1.rds.amazonaws.com/us-gov-west-1/us-gov-west-1-bundle.pem
```

Ensuite, mettez à jour vos applications pour utiliser le nouveau lot de certificats. Le nouveau bundle CA contient à la fois l'ancien certificat CA et le nouveau certificat CA (`rds-ca-rsa4096-g1.pem`). Le fait d'avoir les deux certificats d'autorité de certification dans le nouveau lot de l'autorité de certification vous permet de mettre à jour votre application et votre cluster en deux étapes.

Tout téléchargement du bundle de certificats CA après le 21 décembre 2021 doit utiliser le nouveau bundle de certificats CA. Pour vérifier que votre application utilise le dernier lot de certificats de l'autorité de certification, veuillez consulter Comment puis-je être sûr d'utiliser le tout dernier pack CA ?. Si vous utilisez déjà le dernier lot de certificats de l'autorité de certification dans votre application, vous pouvez passer à l'étape 2.

Pour obtenir des exemples d'utilisation d'une offre groupée CA avec votre application, consultez Chiffrement des données en transit et Connexion avec TLS activé.

> **Note**
>
> Actuellement, le pilote MongoDB Go 1.2.1 accepte uniquement un certificat de serveur d'autorité de certification dans `sslcertificateauthorityfile`. Veuillez consulter

Connexion avec TLS activé pour vous connecter à Amazon DocumentDB à l'aide de Go lorsque TLS est activé.

Étape 2 : Mettre à jour le certificat de serveur

Une fois que l'application a été mise à jour pour utiliser le nouveau bundle CA, l'étape suivante consiste à mettre à jour le certificat du serveur en modifiant chaque instance d'un cluster Amazon DocumentDB. Pour modifier les instances afin qu'elles utilisent le nouveau certificat de serveur, consultez les instructions suivantes.

> **Note**
>
> La mise à jour de vos instances nécessite un redémarrage, ce qui peut entraîner une interruption du service. Vous devez avoir terminé l'Étape 1 avant de mettre à jour le certificat de serveur.

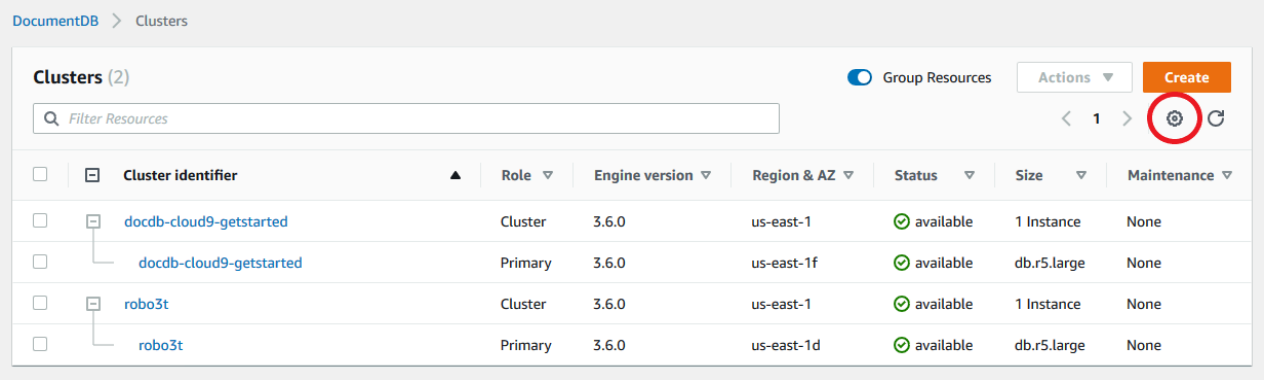
Using the AWS Management Console

Procédez comme suit pour identifier et faire pivoter l'ancien certificat de serveur pour vos instances Amazon DocumentDB existantes à l'aide du. AWS Management Console

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans la liste des régions située dans le coin supérieur droit de l'écran, choisissez celle Région AWS dans laquelle résident vos clusters.
3. wh

   Dans le volet de navigation sur le côté gauche de la console, choisissez Clusters.
4. Vous devrez peut-être identifier les instances qui figurent toujours sur l'ancien certificat de serveur (`rds-ca-2017`). Vous pouvez le faire dans la colonne Autorité de certification qui est masquée par défaut. Pour afficher la colonne Certificate authority (Autorité de certification) procédez comme suit :

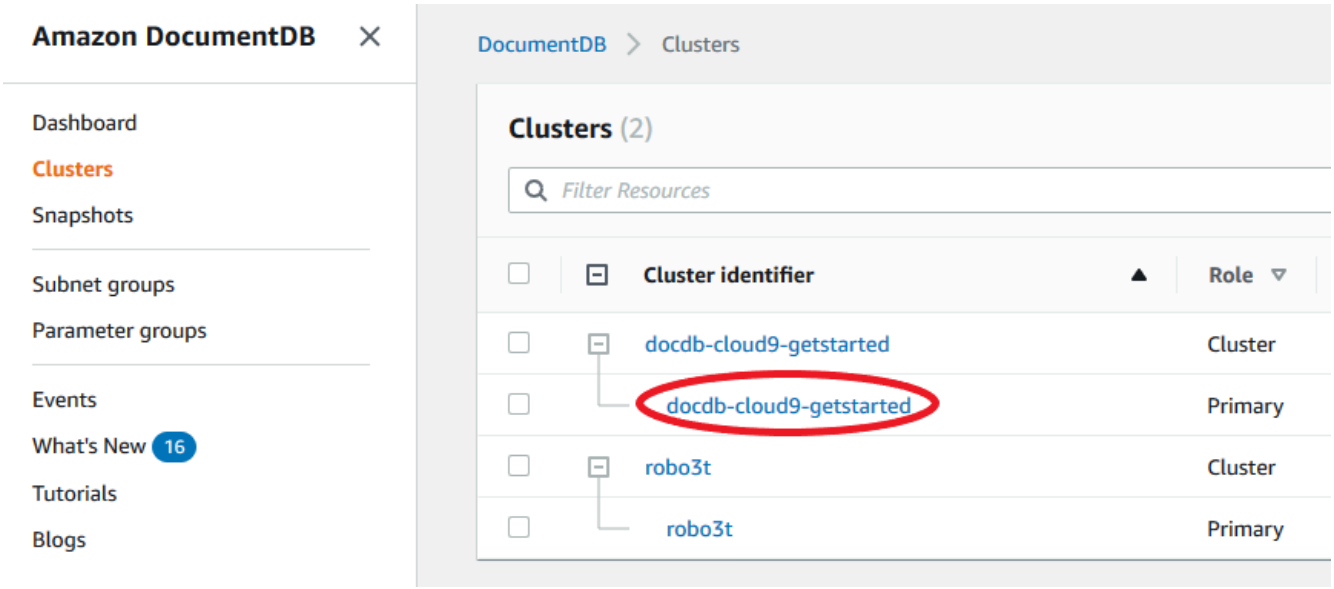
   a. Choisissez l'icône Settings (Paramètres).



b. Dans la liste des colonnes visibles, choisissez la colonne Certificate authority (Autorité de certification) .

c. Choisissez ensuite Confirm (Confirmer) pour enregistrer vos modifications.

5. De retour dans la boîte de navigation Clusters, vous verrez la colonne Cluster Identifier. Vos instances sont répertoriées sous des clusters, comme dans la capture d'écran ci-dessous.



6. Cochez la case située à gauche de l'instance qui vous intéresse.

7. Choisissez Actions, puis Modifier.

8. Sous Autorité de certification, sélectionnez le nouveau certificat de serveur (c.-à-d., `rds-ca-rsa4096-g1`) pour cette instance.

9. Vous pouvez voir un résumé des modifications à la page suivante. Notez qu'il y a une alerte supplémentaire pour vous rappeler de vous assurer que votre application utilise le dernier pack CA de certificats avant de modifier l'instance afin d'éviter d'interrompre la connectivité.

10. Vous pouvez choisir d'appliquer la modification lors de votre prochaine fenêtre de maintenance ou de l'appliquer immédiatement. Si vous avez l'intention de modifier

immédiatement le certificat de serveur, utilisez l'option Apply Immediately (Appliquer immédiatement).

11. Choisissez Modify instance (Modifier l'instance) pour terminer la mise à jour.

Using the AWS CLI

Procédez comme suit pour identifier et faire pivoter l'ancien certificat de serveur pour vos instances Amazon DocumentDB existantes à l'aide du. AWS CLI

1. Pour modifier immédiatement les instances, exécutez la commande suivante pour chaque instance du cluster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa4096-g1 --apply-immediately
```

2. Pour modifier les instances de vos clusters afin d'utiliser le nouveau certificat d'autorité de certification lors de la prochaine fenêtre de maintenance de votre cluster, exécutez la commande suivante pour chaque instance du cluster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa4096-g1 --no-apply-immediately
```

Résolution des problèmes

Si vous rencontrez des problèmes de connexion à votre cluster dans le cadre de la rotation du certificat, nous vous suggérons de procéder comme suit :

- Redémarrez vos instances. La rotation du nouveau certificat nécessite le redémarrage de chacune de vos instances. Si vous avez appliqué le nouveau certificat à une ou plusieurs instances mais que vous ne les avez pas redémarrées, redémarrez vos instances pour appliquer le nouveau certificat. Pour plus d'informations, consultez Redémarrage d'une instance Amazon DocumentDB.
- Vérifiez que vos clients utilisent le dernier ensemble de certificats. veuillez consulter Comment puis-je être sûr d'utiliser le tout dernier pack CA ?.
- Vérifiez que vos instances utilisent le dernier certificat. veuillez consulter Comment savoir laquelle de mes instances Amazon DocumentDB utilise l'ancien ou le nouveau certificat de serveur ?.
- Vérifiez que la dernière autorité de certification est utilisée par votre application. Certains pilotes, comme Java et Go, nécessitent un code supplémentaire pour importer plusieurs certificats à

partir d'un ensemble de certificats vers le magasin de confiance. Pour plus d'informations sur la connexion à Amazon DocumentDB via TLS, consultez. Connexion par programmation à Amazon DocumentDB

- Contactez le support. Si vous avez des questions ou des problèmes, contactez AWS Support.

Questions fréquentes (FAQ)

Voici les réponses à certaines questions courantes concernant les certificats TLS.

Que faire si j'ai des questions ou des problèmes ?

Si vous avez des questions ou des problèmes, contactez AWS Support.

Comment savoir si j'utilise le protocole TLS pour me connecter à mon cluster Amazon DocumentDB ?

Vous pouvez déterminer si votre cluster utilise TLS en examinant le paramètre `tls` du groupe de paramètres de cluster de votre cluster. Si le paramètre `tls` est défini sur `enabled`, vous utilisez le certificat TLS pour vous connecter à votre cluster. Pour plus d'informations, consultez Gestion des groupes de paramètres du cluster Amazon DocumentDB.

Pourquoi mettez-vous à jour les certificats d'autorité de certification et de serveur ?

Les certificats de CA et de serveur Amazon DocumentDB ont été mis à jour dans le cadre des meilleures pratiques de maintenance et de sécurité standard pour Amazon DocumentDB. Les certificats de CA et de serveur actuels expireront le mercredi 18 mai 2022.

Que se passe-t-il si je ne prends aucune mesure avant la date d'expiration ?

Si vous utilisez le protocole TLS pour vous connecter à votre cluster Amazon DocumentDB et que vous n'apportez pas la modification avant le 18 mai 2022, vos applications qui se connectent via TLS ne pourront plus communiquer avec le cluster Amazon DocumentDB.

Amazon DocumentDB ne fera pas automatiquement pivoter vos certificats de base de données avant leur expiration. Vous devez mettre à jour vos applications et vos clusters pour utiliser les nouveaux certificats CA avant ou après la date d'expiration.

Comment savoir laquelle de mes instances Amazon DocumentDB utilise l'ancien ou le nouveau certificat de serveur ?

Pour identifier les instances Amazon DocumentDB qui utilisent toujours l'ancien certificat de serveur, vous pouvez utiliser Amazon AWS Management Console DocumentDB ou le. AWS CLI

En utilisant le AWS Management Console

Pour identifier les instances de vos clusters qui utilisent l'ancien certificat

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans la liste des régions située dans le coin supérieur droit de l'écran, choisissez celle Région AWS dans laquelle résident vos instances.
3. Dans le panneau de navigation situé sur le côté gauche de la console, choisissez Instances.
4. La colonne Certificate authority (Autorité de certification) (masquée par défaut) indique quelles instances se trouvent toujours sur l'ancien certificat de serveur (`rds-ca-2017`) et sur le nouveau (`rds-ca-rsa4096-g1`). Pour afficher la colonne Certificate authority (Autorité de certification) procédez comme suit :
   a. Choisissez l'icône Settings (Paramètres).
   b. Dans la liste des colonnes visibles, choisissez la colonne Certificate authority (Autorité de certification) .
   c. Choisissez ensuite Confirm (Confirmer) pour enregistrer vos modifications.

En utilisant le AWS CLI

Pour identifier les instances de vos clusters qui utilisent l'ancien certificat de serveur, utilisez la commande `describe-db-clusters` avec les éléments suivants.

```
aws docdb describe-db-instances \
    --filters Name=engine,Values=docdb \
    --query 'DBInstances[*].
{CertificateVersion:CACertificateIdentifier,InstanceID:DBInstanceIdentifier}'
```

Comment modifier les instances individuelles de mon cluster Amazon DocumentDB pour mettre à jour le certificat du serveur ?

Nous vous recommandons de mettre à jour simultanément les certificats de serveur pour toutes les instances d'un cluster donné. Pour modifier les instances de votre cluster, vous pouvez utiliser la console ou l' AWS CLI.

> **Note**
>
> La mise à jour de vos instances nécessite un redémarrage, ce qui peut entraîner une interruption du service. Vous devez avoir terminé l'Étape 1 avant de mettre à jour le certificat de serveur.

En utilisant le AWS Management Console

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans la liste des régions située dans le coin supérieur droit de l'écran, choisissez celle Région AWS dans laquelle résident vos clusters.
3. Dans le panneau de navigation situé sur le côté gauche de la console, choisissez Instances.
4. La colonne Certificate authority (Autorité de certification) (masquée par défaut) indique quelles instances se trouvent toujours sur l'ancien certificat du serveur (`rds-ca-2017`). Pour afficher la colonne Certificate authority (Autorité de certification) procédez comme suit :
   a. Choisissez l'icône Settings (Paramètres).
   b. Dans la liste des colonnes visibles, choisissez la colonne Certificate authority (Autorité de certification) .
   c. Choisissez ensuite Confirm (Confirmer) pour enregistrer vos modifications.
5. Sélectionnez une instance à modifier.
6. Choisissez Actions, puis Modifier.
7. Sous Autorité de certification, sélectionnez le nouveau certificat de serveur (`rds-ca-rsa4096-g1`) pour cette instance.
8. Vous pouvez voir un résumé des modifications à la page suivante. Notez qu'il y a une alerte supplémentaire pour vous rappeler de vous assurer que votre application utilise le dernier pack CA de certificats avant de modifier l'instance afin d'éviter d'interrompre la connectivité.

9. Vous pouvez choisir d'appliquer la modification lors de votre prochaine fenêtre de maintenance ou de l'appliquer immédiatement.
10. Choisissez Modify instance (Modifier l'instance) pour terminer la mise à jour.

En utilisant le AWS CLI

Procédez comme suit pour identifier et faire pivoter l'ancien certificat de serveur pour vos instances Amazon DocumentDB existantes à l'aide du. AWS CLI

1. Pour modifier immédiatement les instances, exécutez la commande suivante pour chaque instance du cluster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa4096-g1 --apply-immediately
```

2. Pour modifier les instances de vos clusters afin d'utiliser le nouveau certificat d'autorité de certification lors de la prochaine fenêtre de maintenance de votre cluster, exécutez la commande suivante pour chaque instance du cluster.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa4096-g1 --no-apply-immediately
```

Que se passe-t-il si j'ajoute une nouvelle instance à un cluster existant ?

Toutes les nouvelles instances créées utilisent l'ancien certificat de serveur et nécessitent des connexions TLS à l'aide de l'ancien certificat d'autorité de certification. Toutes les nouvelles instances Amazon DocumentDB créées après le 21 mars 2022 utiliseront par défaut les nouveaux certificats.

Que se passe-t-il s'il y a un remplacement d'instance ou un basculement sur incident sur mon cluster ?

S'il y a un remplacement d'instance dans votre cluster, la nouvelle instance créée continue d'utiliser le même certificat de serveur que celui que l'instance utilisait précédemment. Nous vous recommandons de mettre à jour les certificats de serveur pour toutes les instances en même temps. Si un basculement se produit dans le cluster, le certificat de serveur sur le nouveau serveur principal est utilisé.

Si je n'utilise pas TLS pour me connecter à mon cluster, dois-je toujours mettre à jour chacune de mes instances ?

Si vous n'utilisez pas le protocole TLS pour vous connecter à vos clusters Amazon DocumentDB, aucune action n'est nécessaire.

Si je n'utilise pas TLS pour me connecter à mon cluster mais que je prévois de le faire à l'avenir, que dois-je faire ?

Si vous avez créé un cluster avant le 21 mars 2022, suivez les étapes 1 et 2 de la section précédente pour vous assurer que votre application utilise le bundle CA mis à jour et que chaque instance Amazon DocumentDB utilise le dernier certificat de serveur. Si vous créez un cluster après le 21 mars 2022, celui-ci disposera déjà du dernier certificat de serveur. Pour vérifier que votre application utilise le dernier ensemble de certificats d'autorité de certification, consultez Si je n'utilise pas TLS pour me connecter à mon cluster, dois-je toujours mettre à jour chacune de mes instances ?.

La date limite peut-elle être prolongée au-delà du 18 mai 2022 ?

Si vos candidatures se connectent via TLS, la date limite ne peut pas être prolongée au-delà du 18 mai 2022.

Comment puis-je être sûr d'utiliser le tout dernier pack CA ?

Pour des raisons de compatibilité, les anciens et les nouveaux fichiers groupés CA se nomment `us-gov-west-1-bundle.pem`. Vous pouvez également utiliser des outils comme `openssl` ou `keytool` pour inspecter le lot de l'autorité de certification.

Pourquoi est-ce que je vois « RDS » dans le nom du groupe CA ?

Pour certaines fonctionnalités de gestion, telles que la gestion des certificats, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon Relational Database Service (Amazon RDS).

Si j'ai appliqué le nouveau certificat de serveur, puis-je revenir à l'ancien certificat ?

Si vous devez rétablir une instance à l'ancien certificat de serveur, nous vous recommandons de le faire pour toutes les instances du cluster. Vous pouvez rétablir le certificat de serveur pour chaque instance d'un cluster en utilisant le AWS Management Console ou le AWS CLI.

En utilisant le AWS Management Console

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans la liste des régions située dans le coin supérieur droit de l'écran, choisissez celle Région AWS dans laquelle résident vos clusters.
3. Dans le panneau de navigation situé sur le côté gauche de la console, choisissez Instances.
4. Sélectionnez une instance à modifier. Choisissez Actions, puis Modify (Modifier).
5. Sous Autorité de certification, vous pouvez sélectionner l'ancien certificat de serveur (`rds-ca-2017`).
6. Sélectionnez Continuer pour afficher un résumé de vos modifications.
7. Dans cette page, vous pouvez choisir de planifier l'application de vos modifications dans la prochaine fenêtre de maintenance ou d'appliquer vos modifications immédiatement. Effectuez votre sélection et choisissez Modify instance (Modifier l'instance).

> **Note**
>
> Si vous choisissez d'appliquer les modifications immédiatement, les modifications placées dans la file d'attente des modifications en attente sont également appliquées. Si des modifications en attente ont besoin d'un temps d'arrêt, choisir de les appliquer immédiatement peut entraîner un temps d'arrêt imprévu.

En utilisant le AWS CLI

```
aws docdb modify-db-instance --db-instance-identifier <db_instance_name> ca-
certificate-identifier rds-ca-2017 <--apply-immediately | --no-apply-immediately>
```

Si vous choisissez `--no-apply-immediately`, les modifications seront appliquées lors de la prochaine fenêtre de maintenance du cluster.

Si nous effectuons la restauration à partir d'un instantané ou d'un instant dans le passé, aura-t-il le nouveau certificat de serveur ?

Si vous restaurez un instantané ou si vous effectuez une point-in-time restauration après le 21 mars 2022, le nouveau cluster créé utilisera le nouveau certificat CA.

Que faire si je rencontre des problèmes pour me connecter directement à mon cluster Amazon DocumentDB depuis Mac OS X Catalina ?

Mac OS X Catalina a mis à jour les exigences pour les certificats de confiance. Les certificats fiables doivent désormais être valides pendant 825 jours ou moins (voirhttps://support.apple.com/en-us/HT210176). Les certificats d'instance Amazon DocumentDB sont valides pendant plus de quatre ans, soit plus que le maximum autorisé pour Mac OS X. Pour vous connecter directement à un cluster Amazon DocumentDB depuis un ordinateur exécutant Mac OS X Catalina, vous devez autoriser les certificats non valides lors de la création de la connexion TLS. Dans ce cas, les certificats non valides signifient que la période de validité est supérieure à 825 jours. Vous devez comprendre les risques avant d'autoriser des certificats non valides lors de la connexion à votre cluster Amazon DocumentDB.

Pour vous connecter à un cluster Amazon DocumentDB depuis OS X Catalina à l'aide du paramètre AWS CLI, utilisez le paramètre. `tlsAllowInvalidCertificates`

```
mongo --tls --host <hostname> --username <username> --password <password> --port 27017
 --tlsAllowInvalidCertificates
```

Validation de conformité dans Amazon DocumentDB

La sécurité et la conformité d'Amazon DocumentDB (avec compatibilité MongoDB) sont évaluées par des auditeurs tiers dans le cadre de plusieurs programmes deAWS conformité, notamment :

- Contrôles du système et de l'organisation (SOC) 1, 2 et 3. Pour de plus amples informations, consultez SOC.
- Norme de sécurité des données de l'industrie des cartes de paiement (PCI DSS). Pour de plus amples informations, consultez PCI DSS.
- ISO 9001, 27001, 27017 et 27018. Pour plus d'informations, consultez Certifié ISO.
- Loi sur la transférabilité et l'imputabilité de l'assurance maladie Accord de partenariat (HIPAA BAA). Pour de plus amples informations, consultez Conformité à la loi HIPAA

AWS fournit une liste fréquemment mise à jour des services AWS concernés par les différents programmes de conformité sur la page Services AWS concernés par le programme de conformité.

Les rapports d'audit tiers sont disponibles au téléchargement à l'aide de AWS Artifact. Pour plus d'informations, consultez Téléchargement des rapports dans AWS Artifact.

Pour de plus amples informations sur les programmes de conformité AWS, veuillez consulter Programmes de conformité AWS.

Votre responsabilité en lien avec la conformité lors de l'utilisation d'Amazon DocumentDB est déterminée par la sensibilité de vos données, les objectifs de conformité de votre organisation, ainsi que par la législation et la réglementation applicables. Si votre utilisation d'Amazon DocumentDB est soumise à la conformité à des normes telles que HIPAA ou PCI,AWS fournit des ressources pour vous aider :

- Ressources de conformité AWS – Ensemble de manuels et de guides susceptibles de s'appliquer à votre secteur et à votre emplacement.
- Guides de démarrage rapide de la sécurité et de la conformité – Guides de déploiement qui proposent des considérations architecturales et fournissent des procédures pour déployer des environnements de référence centrés sur la sécurité et la conformité sur AWS.
- AWSConfig : service qui permet d'évaluer comment les configurations de vos ressources se conforment aux pratiques internes, aux normes et aux directives industrielles.
- Security Hub AWS – Vue complète de l'état de votre sécurité au sein d'AWS, qui vous permet de vérifier votre conformité aux normes du secteur et aux bonnes pratiques de sécurité.
- Livre blanc sur l'architecture pour la sécurité et la conformité HIPAA— Livre blanc qui décrit comment les entreprises peuvent utiliserAWS pour créer des applications conformes à la loi HIPAA.

Résilience dans Amazon DocumentDB

L'infrastructure mondiale d'AWS est construite autour de zones de disponibilité et de Régions AWS. Les Régions AWSfournissent plusieurs zones de disponibilité physiquement séparées et isolées, reliées par un réseau à latence faible, à débit élevé et à forte redondance. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone de disponibilité à l'autre sans interruption. Les zones de disponibilité sont plus hautement disponibles, tolérantes aux pannes et évolutives que les infrastructures traditionnelles à un ou plusieurs centres de données.

Un cluster Amazon DocumentDB ne peut être créé que dans un Amazon VPC comportant au moins deux sous-réseaux dans deux zones de disponibilité au moins. En répartissant vos instances de cluster sur deux zones de disponibilité au moins, Amazon DocumentDB garantit que des instances seront disponibles dans votre cluster, dans l'éventualité peu probable d'une défaillance d'une zone de

disponibilité. Le volume de cluster de votre cluster Amazon DocumentDB couvre toujours trois zones de disponibilité afin d'offrir un stockage durable avec un risque moindre de perte des données.

Pour plus d'informations sur les Régions AWS et les zones de disponibilité, consultez Infrastructure mondiale d'AWS.

En plus duAWSInfrastructure mondiale, Amazon DocumentDB propose plusieurs fonctionnalités qui contribuent à la prise en charge des vos besoins en matière de résilience et de sauvegarde de données.

Stockage tolérant aux pannes avec fonction d'autoréparation

Chaque portion de 10 Go de votre volume de stockage est répliquée six fois dans trois zones de disponibilité. Amazon DocumentDB utilise un stockage tolérant les pannes qui gère de manière transparente la perte de deux copies de données au maximum sans affecter la disponibilité en écriture de base de données, et jusqu'à trois copies sans affecter la disponibilité en lecture. Le stockage Amazon DocumentDB est également auto-réparateur ; les blocs de données et les disques sont analysés en continu pour détecter les erreurs et remplacés automatiquement.

Sauvegardes et restauration manuelles

Amazon DocumentDB permet de créer des sauvegardes complètes de votre cluster pour une rétention et une restauration à long terme. Pour plus d'informations, consultez Sauvegarde et restauration dans Amazon DocumentDB.

Restauration à un instant dans le passé

La restauration à un instant dans le passé permet Amazon DocumentDB contre les opérations d'écriture ou de suppression accidentelles. Grâce à la restauration à un instant dans le passé, vous n'avez plus à vous soucier de la création, de la maintenance ou de la planification des sauvegardes à la demande. Pour de plus amples informations, veuillez consulter Restaurez à un instant dans le passé.

Sécurité de l'infrastructure dans Amazon DocumentDB

En tant que service géré, Amazon DocumentDB est protégé parAWSsécurité du réseau mondial. Pour plus d'informations sur les services de sécurité AWS et la manière dont AWS protège l'infrastructure, consultez la section Sécurité du cloud AWS. Pour concevoir votre environnement AWS en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la

section Protection de l'infrastructure dans le Security Pillar AWS Well-Architected Framework (Pilier de sécurité de l'infrastructure Well-Architected Framework).

Tu utilisesAWSappels d'API publiés pour accéder à Amazon DocumentDB via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et nous 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.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Vous pouvez appeler ces opérations d'API à partir de n'importe quel endroit du réseau. Vous pouvez utiliser les politiques Amazon DocumentDB pour contrôler l'accès à partir de points de terminaison Amazon Virtual Private Cloud (Amazon VPC) ou de VPC spécifiques. En fait, cela isole l'accès réseau à une ressource Amazon DocumentDB donnée uniquement du VPC spécifique au sein duAWSréseau.

> **Note**
>
> Amazon DocumentDB ne prend pas en charge les politiques d'accès basées sur les ressources.

Bonnes pratiques de sécurité pour Amazon DocumentDB

Pour obtenir les meilleures pratiques de sécurité, vous devez utiliserAWS Identity and Access ManagementUtilisez des comptes (IAM) pour contrôler l'accès aux opérations d'API Amazon DocumentDB, particulièrement les opérations qui créent, modifient ou suppriment des ressources Amazon DocumentDB. Les ressources de ce type incluent les clusters, les groupes de sécurité et les groupes de paramètres. Utilisez également IAM pour contrôler les actions qui effectuent des tâches administratives courantes telles que la sauvegarde et la restauration de clusters. Lorsque vous créez des rôles IAM, utilisez le principe du moindre privilège.

- Appliquez le principe du moindre privilège avec le contrôle d'accès basé sur les rôles.
- Attribuez un compte IAM individuel à chaque personne qui gère des ressources Amazon DocumentDB. N'utilisez pas leCompte AWSutilisateur racine pour gérer les ressources Amazon DocumentDB. Créez un utilisateur IAM pour chaque personne, y compris vous-même.
- Accordez à chaque utilisateur l'ensemble minimum d'autorisations requises pour exécuter ses tâches.
- Utilisez des groupes IAM pour gérer efficacement des autorisations pour plusieurs utilisateurs. Pour de plus amples informations sur IAM, veuillez consulter le Guide de l'utilisateur IAM. Pour de plus amples informations sur les bonnes pratiques IAM, veuillez consulter Bonnes pratiques IAM.
- Effectuez une rotation régulière des informations d'identification IAM.
- ConfigurationAWSUtilisez Secrets Manager pour effectuer une rotation automatique des secrets pour Amazon DocumentDB. Pour de plus amples informations, veuillez consulterRotation de votreAWSSecrets de Secrets ManageretRotation des secrets pour Amazon DocumentDBdans leAWSGuide d'utilisation de Secrets Manager.
- Utilisez le protocole TLS (Transport Layer Security) et le chiffrement au repos pour chiffrer vos données.

Audit des événements Amazon DocumentDB

Avec Amazon DocumentDB (compatible avec MongoDB), vous pouvez auditer les événements qui ont été effectués dans votre cluster. Les exemples d'événements enregistrés incluent les tentatives d'authentification réussies et celles ayant échoué, la suppression d'une collection dans une base de données ou la création d'un index. Par défaut, l'audit est désactivé sur Amazon DocumentDB et nécessite que vous acceptiez d'utiliser cette fonctionnalité.

Lorsque l'audit est activé, Amazon DocumentDB enregistre les événements relatifs au langage de définition des données (DDL), au langage de manipulation des données (DML), à l'authentification, à l'autorisation et aux événements de gestion des utilisateurs dans Amazon Logs. CloudWatch Lorsque l'audit est activé, Amazon DocumentDB exporte les enregistrements d'audit de votre cluster (documents JSON) vers Amazon CloudWatch Logs. Vous pouvez utiliser Amazon CloudWatch Logs pour analyser, surveiller et archiver vos événements d'audit Amazon DocumentDB.

Bien qu'Amazon DocumentDB ne facture aucun coût supplémentaire pour activer l'audit, des tarifs standard vous sont facturés pour l'utilisation des CloudWatch journaux. Pour plus d'informations sur la tarification des CloudWatch journaux, consultez CloudWatch les tarifs Amazon.

La fonctionnalité d'audit Amazon DocumentDB est nettement différente de l'utilisation des ressources de service surveillée avec. AWS CloudTrail CloudTrail enregistre les opérations effectuées avec le AWS Command Line Interface (AWS CLI) ou AWS Management Console sur des ressources telles que des clusters, des instances, des groupes de paramètres et des instantanés. L'audit des AWS ressources CloudTrail est activé par défaut et ne peut pas être désactivé. La fonctionnalité d'audit Amazon DocumentDB est une fonctionnalité optionnelle. Elle enregistre les opérations qui ont lieu au sein de votre cluster sur des objets, par exemple sur des bases de données, des collections, des index et des utilisateurs.

Rubriques

Événements pris en charge

L'audit Amazon DocumentDB prend en charge les catégories d'événements suivantes :

- Langage de définition des données (DDL) : inclut les opérations de gestion de base de données, les connexions, la gestion des utilisateurs et les autorisations.
- Événements de lecture du langage de manipulation des données (lectures DML) : incluent `find()` les différents opérateurs d'agrégation, les opérateurs arithmétiques, les opérateurs booléens et les autres opérateurs de requête de lecture.
- Événements d'écriture du langage de manipulation de données (écritures DML) : inclusions `insert(), update(), delete(),` et opérateurs `bulkWrite()`

Les types d'événements sont les suivants.

Type d'événement	Catégorie	Description
`authCheck`	Autorisation	Code de résultat 0 : Succès
		Code de résultat 13 : Tentatives non

Type d'événement	Catégorie	Description
		autorisées d'exécution d'une opération.
`authenticate`	Connexion	Tentatives d'authentification réussies ou en échec sur une nouvelle connexion.
`createDatabase`	DDL	Création d'une nouvelle base de données.
`createCollection`	DDL	Création d'une nouvelle collection dans une base de données.
`createIndex`	DDL	Création d'un nouvel index dans une collection.
`dropCollection`	DDL	Suppression d'une collection dans une base de données.
`dropDatabase`	DDL	Suppression d'une base de données.
`dropIndex`	DDL	Suppression d'un index dans une collection.
`modifyChangeStreams`	DDL	Un flux de modifications a été créé.

Type d'événement	Catégorie	Description
`renameCollection`	DDL	Modification du nom d'une collection au sein d'une base de données.
`createRole`	Gestion des rôles	Création d'un rôle.
`dropAllRolesFromDatabase`	Gestion des rôles	Suppression de tous les rôles dans une base de données.
`dropRole`	Gestion des rôles	Supprimer un rôle.
`grantPrivilegesToRole`	Gestion des rôles	Accorder des privilèges à un rôle.
`grantRolesToRole`	Gestion des rôles	Attribution de rôles à un rôle défini par l'utilisateur.
`revokePrivilegesFromRole`	Gestion des rôles	Révocation des privilèges d'un rôle.
`revokeRolesFromRole`	Gestion des rôles	Révocation des rôles d'un rôle défini par l'utilisateur.
`updateRole`	Gestion des rôles	Mettre à jour un rôle.
`createUser`	Gestion des utilisateurs	Création d'un nouvel utilisateur.
`dropAllUsersFromDatabase`	Gestion des utilisateurs	Suppression de tous les utilisateurs dans une base de données.

Type d'événement	Catégorie	Description
`dropUser`	Gestion des utilisateurs	Suppression d'un utilisateur existant.
`grantRolesToUser`	Gestion des utilisateurs	Attribution de rôles à un utilisateur.
`revokeRolesFromUser`	Gestion des utilisateurs	Révocation des rôles d'un utilisateur.
`updateUser`	UserManagement	Mise à jour d'un utilisateur existant.
`insert`	écriture DML	Insère un ou plusieurs documents dans une collection.
`delete`	écriture DML	Supprime un ou plusieurs documents d'une collection.
`update`	écriture DML	Modifie un ou plusieurs documents existants d'une collection.
`bulkWrite`	écriture DML	Effectue plusieurs opérations d'écriture en contrôlant l'ordre d'exécution.
`count`	Lecture DML	Renvoie le nombre de documents susceptibles de correspondre à une requête find () pour la collection ou la vue.

Type d'événement	Catégorie	Description
`countDocuments`	Lecture DML	Renvoie le nombre de documents correspondant à la requête pour une collection ou une vue.
`find`	Lecture DML	Sélectionne les documents d'une collection ou d'une vue et renvoie le curseur sur les documents sélectionnés.
`findAndModify`	Lecture et écriture DML	Modifie et renvoie un seul document.
`findOneAndDelete`	Lecture et écriture DML	Supprime un seul document en fonction du filtre et des critères de tri, renvoyant le document supprimé.
`findOneAndReplace`	Lecture et écriture DML	Remplace un seul document en fonction du filtre spécifié.
`findOneAndUpdate`	Lecture et écriture DML	Met à jour un seul document en fonction du filtre et des critères de tri.
`aggregate`	Lecture et écriture DML	Prend en charge les API dans le pipeline d'agrégation.

Type d'événement	Catégorie	Description
`distinct`	Lecture DML	Recherche les valeurs distinctes d'un champ spécifié dans une collection ou une vue unique et renvoie les résultats sous forme de tableau.

> **Note**
>
> Les valeurs du champ de paramètre du document d'événement DML sont limitées à 1 Ko. Amazon DocumentDB tronque la valeur si elle dépasse 1 Ko.

> **Note**
>
> Les événements de suppression TTL ne sont pas audités pour le moment.

Activation de l'audit

L'activation de l'audit sur un cluster est un processus en deux étapes. Assurez-vous que les deux étapes sont terminées, sinon les journaux d'audit ne seront pas envoyés à CloudWatch Logs.

Étape 1. Activer le paramètre de cluster audit_logs

Pour activer l'audit, vous devez modifier le `audit_logs` paramètre dans le groupe de paramètres. `audit_logs`est une liste d'événements à consigner, séparés par des virgules. Les événements doivent être spécifiés en minuscules et il ne doit y avoir aucun espace entre les éléments de la liste.

Vous pouvez définir les valeurs suivantes pour le groupe de paramètres :

Valeur	Description	
`ddl`	Cette configuration permettra d'auditer	

Valeur	Description	
	les événements DDL tels que CreateDatabase, DropDatabase, CreateCollection, DropCollection, CreateIndex, DropIndex, AuthCheck, authenticate, CreateUser, DropUser, User, User, UpdateUser et grantRolesTo revokeRolesFrom dropAllUsersFromDatabase	
`dml_read`	Cette configuration permettra d'auditer les événements de lecture DML tels que find, sort count, distinct, group, projecta, unwind, GeoNear, GeoIntersects, GeoWithin et d'autres opérateurs de requête de lecture MongoDB.	
`dml_write`	Cette configuration permettra d'auditer les événements d'écriture DML tels que insert (), update (), delete () et BulkWrite ()	

Valeur	Description	
`all`	Cette configuration permettra d'auditer les événements de votre base de données, tels que les requêtes de lecture, les requêtes d'écriture, les actions de base de données et les actions d'administrateur.	
`none`	Cette configuration désactivera l'audit	

Valeur	Description	
`enabled` (hérité)	Il s'agit d'un ancien paramètre équivalent à « ddl ». Cette configuration permettra d'auditer les événements DDL tels que CreateDatabase, DropDatabase, CreateCollection, DropCollection, CreateIndex, DropIndex, AuthCheck, authenticate, CreateUser, DropUser, User, User, User, UpdateUser et. grantRolesTo revokeRolesFrom dropAllUsers FromDatabase Nous vous déconseillons d'utiliser ce paramètre car il s'agit d'un ancien paramètre.	
`disabled` (héritage)	Il s'agit d'un ancien paramètre équivalent à « aucun ». Nous vous déconseillons d'utiliser ce paramètre car il s'agit d'un ancien paramètre.	

> **Note**
>
> La valeur par défaut du paramètre de cluster audit_logs est none (legacy "`disabled`«).

Vous pouvez également utiliser les valeurs mentionnées ci-dessus dans des combinaisons.

Valeur	Description
`ddl, dml_read`	Cette configuration permettra d'auditer les événements DDL et les événements de lecture DML.
`ddl, dml_write`	Cette configuration activera l'audit pour les événements DDL et l'écriture DML
`dml_read, dml_write`	Cette configuration activera l'audit de tous les événements DML

> **Note**
>
> Vous ne pouvez pas modifier un groupe de paramètres par défaut.

Pour plus d'informations, consultez les ressources suivantes :

- Création de groupes de paramètres de cluster Amazon DocumentDB

  Après avoir créé un groupe de paramètres, modifiez-le en remplaçant la valeur du paramètre `audit_logs` par `enabled`.
- Modification des groupes de paramètres du cluster Amazon DocumentDB

Étape 2. Activer Amazon CloudWatch Logs Export

Lorsque la valeur du paramètre de `audit_logs` cluster est`enabled`,, ou `ddl` `dml_read`ou `dml_write`, vous devez également activer Amazon DocumentDB pour exporter les journaux vers Amazon. CloudWatch Si vous omettez l'une de ces étapes, les journaux d'audit ne seront pas envoyés à CloudWatch.

Lorsque vous créez un cluster, effectuez un point-in-time-restore instantané ou restaurez un instantané, vous pouvez activer CloudWatch les journaux en suivant ces étapes.

Using the AWS Management Console

Pour permettre à Amazon DocumentDB d'exporter des journaux à CloudWatch l'aide de la console, consultez les rubriques suivantes :

- Lors de la création d'un cluster — DansCréation d'un cluster et d'une instance principale à l'aide du AWS Management Console, voir Créer un cluster : configurations supplémentaires (étape 5, Exportations du journal)
- Lors de la modification d'un cluster existant : Modification d'un cluster Amazon DocumentDB
- Lorsque vous effectuez une restauration instantanée d'un cluster : Restauration d'un cluster à partir d'un instantané
- Lorsque vous effectuez une point-in-time restauration : Restaurez à un instant dans le passé

Using the AWS CLI

Pour activer les journaux d'audit lors de la création d'un cluster

Le code suivant crée le cluster `sample-cluster` et active les journaux CloudWatch d'audit.

Example

Pour Linux, macOS ou Unix :

```
aws docdb create-db-cluster \
    --db-cluster-identifier sample-cluster \
    --port 27017 \
    --engine docdb \
    --master-username master-username \
    --master-user-password password \
    --db-subnet-group-name default \
```

```
    --enable-cloudwatch-logs-exports audit
```

Pour Windows :

```
aws docdb create-db-cluster ^
    --db-cluster-identifier sample-cluster ^
    --port 27017 ^
    --engine docdb ^
    --master-username master-username ^
    --master-user-password password ^
    --db-subnet-group-name default ^
    --enable-cloudwatch-logs-exports audit
```

Pour activer les journaux d'audit lors de la modification d'un cluster existant

Le code suivant modifie le cluster `sample-cluster` et active les journaux CloudWatch d'audit.

Example

Pour Linux, macOS ou Unix :

```
aws docdb modify-db-cluster \
   --db-cluster-identifier sample-cluster \
   --cloudwatch-logs-export-configuration '{"EnableLogTypes":["audit"]}'
```

Pour Windows :

```
aws docdb modify-db-cluster ^
   --db-cluster-identifier sample-cluster ^
   --cloudwatch-logs-export-configuration '{"EnableLogTypes":["audit"]}'
```

Le résultat de ces opérations ressemble à ceci (format JSON).

```
{
    "DBCluster": {
        "HostedZoneId": "ZNKXH85TT8WVW",
        "StorageEncrypted": false,
        "DBClusterParameterGroup": "default.docdb4.0",
        "MasterUsername": "<user-name>",
        "BackupRetentionPeriod": 1,
        "Port": 27017,
        "VpcSecurityGroups": [
```

```
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "DBClusterArn": "arn:aws:rds:us-east-1:900083794985:cluster:sample-cluster",
        "Status": "creating",
        "Engine": "docdb",
        "EngineVersion": "4.0.0",
        "MultiAZ": false,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1f"
        ],
        "DBSubnetGroup": "default",
        "DBClusterMembers": [],
        "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "PreferredMaintenanceWindow": "wed:03:08-wed:03:38",
        "AssociatedRoles": [],
        "ClusterCreateTime": "2019-02-13T16:35:04.756Z",
        "DbClusterResourceId": "cluster-YOS52CUXGDTNKDQ7DH72I4LED4",
        "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
        "PreferredBackupWindow": "07:16-07:46",
        "DBClusterIdentifier": "sample-cluster"
    }
}
```

Désactivation de l'audit

Vous pouvez désactiver l'audit en désactivant l'exportation CloudWatch des journaux et en désactivant le `audit_logs` paramètre.

Désactivation de l'exportation CloudWatch des journaux

Vous pouvez désactiver l'exportation des journaux d'audit à l'aide de la AWS Management Console ou de l'AWS CLI.

Using the AWS Management Console

La procédure suivante utilise le AWS Management Console pour désactiver l'exportation des journaux vers Amazon DocumentDB vers. CloudWatch

Pour désactiver les journaux d'audit

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters. Choisissez ensuite le bouton à gauche du nom du cluster pour lequel vous souhaitez désactiver l'exportation des journaux.
3. Choisissez Actions, puis Modify (Modifier).
4. Faites défiler jusqu'à la section Log exports (Exportations de journaux), puis choisissez Disabled (Désactivé).
5. Choisissez Continue (Continuer).
6. Vérifiez vos modifications, puis choisissez quand cette modification devra être appliquée à votre cluster.
   - Appliquer pendant la fenêtre de maintenance planifiée suivante
   - Appliquer immédiatement
7. Choisissez Modifier le cluster.

Using the AWS CLI

Le code suivant modifie le cluster `sample-cluster` et désactive les journaux CloudWatch d'audit.

Example

Pour Linux, macOS ou Unix :

```
aws docdb modify-db-cluster \
   --db-cluster-identifier sample-cluster \
   --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit"]}'
```

Pour Windows :

```
aws docdb modify-db-cluster ^
```

```
    --db-cluster-identifier sample-cluster ^
    --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit"]}'
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "DBCluster": {
        "DBClusterParameterGroup": "default.docdb4.0",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "MasterUsername": "<user-name>",
        "Status": "available",
        "Engine": "docdb",
        "Port": 27017,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1f"
        ],
        "EarliestRestorableTime": "2019-02-13T16:35:50.387Z",
        "DBSubnetGroup": "default",
        "LatestRestorableTime": "2019-02-13T16:35:50.387Z",
        "DBClusterArn": "arn:aws:rds:us-east-1:900083794985:cluster:sample-
cluster2",
        "Endpoint": "sample-cluster2.cluster-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
        "ReaderEndpoint": "sample-cluster2.cluster-ro-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
        "BackupRetentionPeriod": 1,
        "EngineVersion": "4.0.0",
        "MultiAZ": false,
        "ClusterCreateTime": "2019-02-13T16:35:04.756Z",
        "DBClusterIdentifier": "sample-cluster2",
        "AssociatedRoles": [],
        "PreferredBackupWindow": "07:16-07:46",
        "DbClusterResourceId": "cluster-YOS52CUXGDTNKDQ7DH72I4LED4",
        "StorageEncrypted": false,
        "PreferredMaintenanceWindow": "wed:03:08-wed:03:38",
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
```

```
        ]
    }
}
```

Désactivation du paramètre audit_logs

Pour désactiver le paramètre `audit_logs` de votre cluster, vous pouvez modifier ce dernier de façon à ce qu'il utilise un groupe de paramètres dans lequel la valeur du paramètre `audit_logs` est `disabled`. Vous pouvez également modifier la valeur du paramètre `audit_logs` dans le groupe de paramètres du cluster afin qu'elle soit `disabled`.

Pour plus d'informations, consultez les rubriques suivantes :

- Modification d'un cluster Amazon DocumentDB
- Modification des groupes de paramètres du cluster Amazon DocumentDB

Accès à vos événements d'audit

Suivez les étapes ci-dessous pour accéder à vos événements d'audit sur Amazon CloudWatch.

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Assurez-vous que vous vous trouvez dans la même région que votre cluster Amazon DocumentDB.
3. Dans le panneau de navigation, sélectionnez Logs (Journaux).
4. Pour rechercher les journaux d'audit de votre cluster, dans la liste, trouvez et choisissez **`/aws/docdb/`*`yourClusterName`*`/audit`**.

   Les événements d'audit pour chacune de vos instances sont disponibles sous chacun des noms d'instance respectifs.

Sauvegarde et restauration dans Amazon DocumentDB

Amazon DocumentDB (compatible avec MongoDB) sauvegarde en permanence vos données sur Amazon Simple Storage Service (Amazon S3) pendant 1 à 35 jours afin que vous puissiez les restaurer rapidement à tout moment pendant la période de conservation des sauvegardes. Amazon DocumentDB prend également des instantanés automatiques de vos données dans le cadre de ce processus de sauvegarde continue.

> **Note**
>
> Il s'agit de compartiments Amazon S3 gérés par des services et vous n'aurez pas accès aux fichiers de sauvegarde. Si vous souhaitez contrôler vos propres sauvegardes, suivez les instructions relatives au dumping, à la restauration, à l'importation et à l'exportation de données.

Vous pouvez également conserver les données de sauvegarde au-delà de la période de rétention des sauvegardes, en créant un instantané manuel des données de votre cluster. Le processus de sauvegarde n'a aucune incidence sur les performances de votre cluster.

Cette section décrit les cas d'utilisation des fonctionnalités de sauvegarde d'Amazon DocumentDB et explique comment gérer les sauvegardes de vos clusters Amazon DocumentDB.

Rubriques

- Suppression d'un instantané de cluster

Sauvegarde et restauration : Concepts

Nom	Description	API (verbes)

Période de rétention des sauvegardes	Période comprise entre 1 et 35 jours pendant laquelle vous pouvez effectuer une point-in-time restauration.	`create-db-cluster`
		`modify-db-cluster`
		`restore-db-cluster-to-point-in-time`

Volume de stockage Amazon DocumentDB	Volume de stockage hautement disponible et hautement durable qui réplique les données de six façons dans trois zones de disponibilité. Un cluster	`create-db-cluster` `delete-db-cluster`

Nom	Description	API (verbes)
	Amazon DocumentDB est extrêmement durable quel que soit le nombre d'instances qu'il contient.	

Fenêtre de sauvegarde	Période de la journée pendant laquelle sont pris les instantanés automatiques.	`create-db-cluster`
		`describe-db-cluster`
		`modify-db-cluster`

Nom	Description	API (verbes)

Instantané automatique	Des instantanés quotidiens qui sont des sauvegardes complètes du cluster et sont automatiquement créés par le processus de sauvegarde continue dans Amazon DocumentDB.	`restore-db-cluster-from-snapshot`
		
		`describe-db-cluster-snapshot-attributes`
		
		`describe-db-cluster-snapshots`

Nom	Description	API (verbes)

Instantanés manuels	Instantanés que vous créez manuellement pour conserver les sauvegardes complètes d'un cluster au-delà de la période de sauvegarde.	`create-db-cluster-snapshot`
		`copy-db-cluster-snapshot`
		`delete-db-cluster-snapshot`
		`describe-db-cluster-snapshot-attributes`
		`describe-db-cluster-snapshots`
		`modify-db-cluster-snapshot-attribute`

Présentation de l'utilisation du stockage de sauvegarde

Le stockage de sauvegarde Amazon DocumentDB consiste en des sauvegardes continues pendant la période de conservation des sauvegardes et en des instantanés manuels en dehors de cette période. Pour contrôler votre utilisation du stockage des sauvegardes, vous pouvez réduire l'intervalle de rétention des sauvegardes, supprimer d'anciens instantanés manuels lorsqu'ils ne sont plus nécessaires, ou les deux. Pour obtenir des informations générales sur les sauvegardes Amazon DocumentDB, consultez. Sauvegarde et restauration dans Amazon DocumentDB Pour obtenir des informations sur les tarifs relatifs au stockage de sauvegarde Amazon DocumentDB, consultez la section Tarification d'Amazon DocumentDB.

Pour contrôler vos coûts, vous pouvez surveiller la quantité de stockage consommée par les sauvegardes continues et les instantanés manuels qui persistent au-delà de la période de rétention. Vous pouvez ensuite réduire l'intervalle de rétention des sauvegardes et supprimer des instantanés manuels lorsqu'ils ne sont plus nécessaires.

Vous pouvez utiliser les CloudWatch métriques Amazon `TotalBackupStorageBilledSnapshotStorageUsed`, et

`BackupRetentionPeriodStorageUsed` pour examiner et surveiller la quantité de stockage utilisée par vos sauvegardes Amazon DocumentDB, comme suit :

- `BackupRetentionPeriodStorageUsed` représente la quantité de stockage de sauvegarde utilisée pour stocker les sauvegardes continues pour l'instant. Cette valeur de métrique dépend de la taille du volume de cluster et du nombre de modifications apportées au cours de la période de rétention. Cependant, à des fins de facturation, la métrique ne dépasse pas la taille du volume de cluster cumulée au cours de la période de rétention. Par exemple, si votre cluster est de 100 Gio et que votre période de rétention est de deux jours, la valeur maximale `BackRetentionPeriodStorageUsed` est de 200 Gio (100 Gio +100 Gio).

- `SnapshotStorageUsed` représente la quantité de stockage de sauvegarde utilisée pour stocker les instantanés manuels au-delà de la période de rétention des sauvegardes. Les instantanés manuels effectués pendant la période de rétention ne sont pas pris en compte dans le calcul de votre stockage de sauvegarde. Il en va de même pour les instantanés automatiques. La taille de chaque instantané correspond à la taille du volume de cluster au moment où vous avez pris l'instantané. La valeur de `SnapshotStorageUsed` dépend du nombre d'instantanés que vous conservez et de la taille de chaque instantané. Par exemple, supposons que vous disposez d'un instantané en dehors de la période de rétention et que la taille du volume de cluster était de 100 Gio lorsque l'instantané a été pris. La quantité de `SnapshotStorageUsed` est de 100 Gio.

- `TotalBackupStorageBilled` représente la somme de `BackupRetentionPeriodStorageUsed` et `SnapshotStorageUsed`, moins un volume de stockage de sauvegarde disponible égal à la taille du volume de cluster pour une journée. Par exemple, si la taille de votre cluster est de 100 GiB, que vous avez un jour de rétention et que vous avez un instantané en dehors de la période de rétention, la valeur `TotalBackupStorageBilled` est de 100 GiB (100 GiB + 100 GiB - 100 GiB).

- Ces métriques sont calculées indépendamment pour chaque cluster Amazon DocumentDB.

Vous pouvez surveiller vos clusters Amazon DocumentDB et créer des rapports à l'aide de CloudWatch métriques via la CloudWatch console. Pour plus d'informations sur l'utilisation CloudWatch des métriques, consultezSurveillance Amazon DocumentDB.

Vidage, restauration, importation et exportation de données

Vous pouvez utiliser les `mongoimport` utilitaires`mongodump`, `mongorestore``mongoexport`, et pour déplacer des données vers et depuis votre cluster Amazon DocumentDB. Cette section décrit l'objectif de chacun de ces outils et configurations pour vous aider à obtenir de meilleures performances.

Rubriques

`mongodump`

L'utilitaire `mongodump` crée une sauvegarde binaire (BSON) d'une base de données MongoDB. `mongodump`Cet outil est la méthode préférée pour transférer les données de votre déploiement MongoDB source lorsque vous souhaitez les restaurer dans votre cluster Amazon DocumentDB en raison des économies de taille obtenues en stockant les données dans un format binaire.

En fonction des ressources disponibles sur l'instance ou la machine que vous utilisez pour exécuter la commande, vous pouvez accélérer le processus `mongodump` en augmentant le nombre de connexions parallèles déversées par rapport à la valeur par défaut 1 à l'aide de l'`--numParallelCollections`option. En règle générale, commencez par un travailleur par vCPU sur l'instance principale de votre cluster Amazon DocumentDB.

> **Note**
>
> Nous recommandons les outils de base de données MongoDB jusqu'à la version 100.6.1 incluse pour Amazon DocumentDB. Vous pouvez accéder aux téléchargements des outils de base de données MongoDB ici.

Exemple d'utilisation

Voici un exemple d'utilisation de l'`mongodump`utilitaire dans le cluster Amazon DocumentDB,. `sample-cluster`

```
mongodump --ssl \
    --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \
    --collection=sample-collection \
    --db=sample-database \
    --out=sample-output-file \
    --numParallelCollections 4  \
    --username=sample-user \
    --password=abc0123 \
    --sslCAFile global-bundle.pem
```

`mongorestore`

L'utilitaire `mongorestore` vous permet de restaurer une sauvegarde binaire (BSON) d'une base de données créée avec l'utilitaire `mongodump`. Vous pouvez améliorer les performances de restauration en augmentant le nombre d'agents de travail pour chaque collection pendant la restauration avec l'option `--numInsertionWorkersPerCollection` (la valeur par défaut est 1). En règle générale, commencez par un travailleur par vCPU sur l'instance principale de votre cluster Amazon DocumentDB.

Exemple d'utilisation

Voici un exemple d'utilisation de l'`mongorestore`utilitaire dans le cluster Amazon DocumentDB,. `sample-cluster`

```
mongorestore --ssl \
    --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \
    --username=sample-user \
    --password=abc0123 \
    --sslCAFile global-bundle.pem <fileToBeRestored>
```

`mongoexport`

L'`mongoexport`outil exporte les données d'Amazon DocumentDB aux formats de fichier JSON, CSV ou TSV. L'outil `mongoexport` constitue la méthode privilégiée pour exporter des données qui doivent être lisibles par l'homme ou par la machine.

> **Note**
>
> `mongoexport` ne prend pas directement en charge les exportations parallèles. Toutefois, il est possible d'augmenter les performances en exécutant simultanément plusieurs tâches `mongoexport` pour différentes collections.

Exemple d'utilisation

Voici un exemple d'utilisation de l'`mongoexport`outil dans le cluster Amazon DocumentDB. `sample-cluster`

```
mongoexport --ssl \
    --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \
    --collection=sample-collection \
    --db=sample-database \
    --out=sample-output-file \
    --username=sample-user \
    --password=abc0123 \
    --sslCAFile global-bundle.pem
```

`mongoimport`

L'`mongoimport`outil importe le contenu de fichiers JSON, CSV ou TSV dans un cluster Amazon DocumentDB. Vous pouvez utiliser le paramètre `--numInsertionWorkers` pour paralléliser et accélérer l'importation (la valeur par défaut est 1).

Exemple d'utilisation

Voici un exemple d'utilisation de l'`mongoimport`outil dans le cluster Amazon DocumentDB. `sample-cluster`

```
mongoimport --ssl \
    --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \
    --collection=sample-collection \
    --db=sample-database \
    --file=<yourFile> \
    --numInsertionWorkers 4 \
    --username=sample-user \
```

```
    --password=abc0123 \
    --sslCAFile global-bundle.pem
```

Didacticiel

Le didacticiel suivant explique comment utiliser les `mongoimport` utilitaires`mongodump`, `mongorestore``mongoexport`, et pour déplacer des données vers et depuis un cluster Amazon DocumentDB.

1. Conditions préalables — Avant de commencer, assurez-vous que votre cluster Amazon DocumentDB est provisionné et que vous avez accès à une instance Amazon EC2 dans le même VPC que votre cluster. Pour plus d'informations, consultez Connectez-vous à l'aide d'Amazon EC2.

   Pour pouvoir utiliser les outils utilitaires mongo, le mongodb-org-tools package doit être installé dans votre instance EC2, comme suit.

   ```
   sudo yum install mongodb-org-tools-4.0.18
   ```

   Amazon DocumentDB utilisant le chiffrement TLS (Transport Layer Security) par défaut, vous devez également télécharger le fichier d'autorité de certification (CA) Amazon RDS pour utiliser le shell mongo pour vous connecter, comme suit.

   ```
   wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
   ```

2. Télécharger des exemples de données — Pour ce didacticiel, vous allez télécharger des exemples de données contenant des informations sur les restaurants.

   ```
   wget https://raw.githubusercontent.com/ozlerhakan/mongodb-json-files/master/
   datasets/restaurant.json
   ```

3. Importez les exemples de données dans Amazon DocumentDB : les données étant au format JSON logique, vous utiliserez l'`mongoimport`utilitaire pour les importer dans votre cluster Amazon DocumentDB.

   ```
   mongoimport --ssl \
       --host="tutorialCluster.amazonaws.com:27017" \
       --collection=restaurants \
       --db=business \
       --file=restaurant.json \
   ```

```
    --numInsertionWorkers 4 \
    --username=<yourUsername> \
    --password=<yourPassword> \
    --sslCAFile global-bundle.pem
```

4. Videz les données avec **`mongodump`** — Maintenant que vous avez des données dans votre cluster Amazon DocumentDB, vous pouvez effectuer un vidage binaire de ces données à l'aide de l'`mongodump`utilitaire.

```
mongodump --ssl \
    --host="tutorialCluster.us-east-1.docdb.amazonaws.com:27017"\
    --collection=restaurants \
    --db=business \
    --out=restaurantDump.bson \
    --numParallelCollections 4 \
    --username=<yourUsername> \
    --password=<yourPassword> \
    --sslCAFile global-bundle.pem
```

5. Supprimer la **`restaurants`** collection — Avant de restaurer la `restaurants` collection dans la `business` base de données, vous devez d'abord supprimer la collection qui existe déjà dans cette base de données, comme suit.

```
use business
```

```
db.restaurants.drop()
```

6. Restaurez les données avec **`mongorestore`** — Grâce au vidage binaire des données de l'étape 3, vous pouvez désormais utiliser l'`mongorestore`utilitaire pour restaurer vos données sur votre cluster Amazon DocumentDB.

```
mongorestore --ssl \
    --host="tutorialCluster.us-east-1.docdb.amazonaws.com:27017" \
    --numParallelCollections 4 \
    --username=<yourUsername> \
    --password=<yourPassword> \
    --sslCAFile global-bundle.pem restaurantDump.bson
```

7. Exporter les données à l'aide de `mongoexport` — Pour terminer le didacticiel, exportez les données de votre cluster au format d'un fichier JSON, identique à celui que vous avez importé à l'étape 1.

```
mongoexport --ssl \
    --host="tutorialCluster.node.us-east-1.docdb.amazonaws.com:27017" \
    --collection=restaurants \
    --db=business \
    --out=restaurant2.json \
    --username=<yourUsername> \
    --password=<yourPassword> \
    --sslCAFile global-bundle.pem
```

8. Validation — Vous pouvez vérifier que le résultat de l'étape 5 donne le même résultat que celui de l'étape 1 à l'aide des commandes suivantes.

```
wc -l restaurant.json
```

Sortie de cette commande :

```
2548 restaurant.json
```

```
wc -l restaurant2.json
```

Sortie de cette commande :

```
2548 restaurant2.json
```

Considérations relatives aux instantanés de cluster

Amazon DocumentDB crée des instantanés automatiques quotidiens de votre cluster pendant la fenêtre de sauvegarde de celui-ci. Amazon DocumentDB enregistre les instantanés automatiques de votre cluster conformément à la période de conservation des sauvegardes que vous spécifiez. Le cas échéant, vous pouvez récupérer votre cluster à n'importe quel moment pendant la période de rétention des sauvegardes. Les instantanés automatiques ne sont pas exécutés pendant l'exécution d'une copie dans la même région pour le même cluster.

Rubriques

- Période de conservation de la sauvegarde
- Copier le chiffrement des instantanés du cluster

Outre des instantanés automatiques du cluster, vous pouvez également créer manuellement un instantané du cluster. Vous pouvez copier les instantanés manuels et automatiques. Pour plus d'informations, consultez Création d'un instantané manuel d'un cluster et Copier des instantanés du cluster Amazon DocumentDB.

> **Note**
>
> Votre cluster doit se trouve en état disponible pour prendre un instantané automatique. Vous ne pouvez pas partager un instantané de cluster automatisé Amazon DocumentDB. Pour contourner le problème, vous pouvez créer un instantané manuel en copiant l'instantané automatisé, puis en partageant cette copie. Pour de plus amples informations sur la copie d'un instantané, veuillez consulter Copier des instantanés du cluster Amazon DocumentDB. Pour de plus amples informations sur la restauration d'un cluster à partir d'un instantané, veuillez consulter Restauration d'un cluster à partir d'un instantané.

Stockage de sauvegarde

Votre stockage de sauvegarde Amazon DocumentDB pour chacune Région AWS est composé du stockage de sauvegarde nécessaire à votre période de conservation des sauvegardes, qui inclut les instantanés de cluster automatiques et manuels dans cette région. La période de conservation de la sauvegarde par défaut est de 1 jour. Pour plus d'informations sur la tarification du stockage de sauvegarde, consultez la section Tarification d'Amazon DocumentDB.

Lorsque vous supprimez un cluster, tous ses instantanés automatiques sont supprimés et ne peuvent pas être récupérés. Toutefois, les instantanés manuels ne sont pas supprimés lorsque vous supprimez un cluster. Si vous choisissez de demander à Amazon DocumentDB de créer un instantané final (instantané manuel) avant la suppression de votre cluster, vous pouvez utiliser l'instantané final pour récupérer votre cluster.

Pour de plus amples informations sur les instantanés et le stockage, veuillez consulter Présentation de l'utilisation du stockage de sauvegarde.

Fenêtre de sauvegarde

Les instantanés automatiques sont exécutés chaque jour pendant le créneau de sauvegarde préféré. Si l'instantané a besoin de plus de temps que la durée allouée au créneau de sauvegarde, le processus de sauvegarde se poursuit jusqu'au bout, même si le créneau de sauvegarde est terminé. Le créneau de sauvegarde ne peut pas chevaucher la fenêtre de maintenance hebdomadaire pour le cluster.

Si vous ne spécifiez pas de fenêtre de sauvegarde préférée lorsque vous créez le cluster, Amazon DocumentDB attribue une fenêtre de sauvegarde par défaut de 30 minutes. Ce créneau est choisi de façon aléatoire à partir d'un bloc de 8 heures associées à la région de votre cluster. Vous pouvez modifier votre créneau de sauvegarde préféré en modifiant le cluster. Pour plus d'informations, consultez Modification d'un cluster Amazon DocumentDB.

Nom de la région	Région	Bloc horaire UTC
USA Est (Ohio)	us-east-2	3 h 00-11 h 00
US East (Virginie du Nord)	us-east-1	3 h 00-11 h 00
USA Ouest (Oregon)	us-west-2	6 h 00-14 h 00
Asie-Pacifique (Hong Kong)	ap-east-1	6 h 00-14 h 00
Asie-Pacifique (Hyderabad)	ap-south-2	6 H 30 — 14 H 30
Asie-Pacifique (Mumbai)	ap-south-1	6 h 00-14 h 00
Asie-Pacifique (Séoul)	ap-northeast-2	13 h 00-21 h 00
Asie-Pacifique (Singapour)	ap-southeast-1	14h00-22h00
Asie-Pacifique (Sydney)	ap-southeast-2	12h00-20h00
Asie-Pacifique (Tokyo)	ap-northeast-1	13 h 00-21 h 00
Canada (Centre)	ca-central-1	3 h 00-11 h 00
Chine (Beijing)	cn-north-1	6 h 00-14 h 00
Chine (Ningxia)	cn-northwest-1	6 h 00-14 h 00

Nom de la région	Région	Bloc horaire UTC
Europe (Francfort)	eu-central-1	21 h 00 - 5 h 00
Europe (Irlande)	eu-west-1	22 h 00-6 h 00
Europe (Londres)	eu-west-2	22 h 00-6 h 00
Europe (Milan)	eu-south-1	02:00-10:00
Europe (Paris)	eu-west-3	23:59-07:29
Moyen-Orient (EAU)	me-central-1	05 H 00 — 13 H 00
Amérique du Sud (São Paulo)	sa-east-1	00:00-08:00
AWS GovCloud (USA Est)	us-gov-east-1	17:00-01:00
AWS GovCloud (US-Ouest)	us-gov-west-1	6 h 00-14 h 00

Période de conservation de la sauvegarde

La période de conservation des sauvegardes correspond au nombre de jours pendant lesquels une sauvegarde automatique est conservée avant d'être automatiquement supprimée. Amazon DocumentDB prend en charge une période de conservation des sauvegardes de 1 à 35 jours.

Vous pouvez configurer la période de rétention des sauvegardes lors de la création d'un cluster. Si vous ne définissez pas clairement la période de rétention des sauvegardes, la période de rétention des sauvegardes par défaut de 1 jour est attribuée à votre cluster. Après avoir créé un cluster, vous pouvez modifier la période de conservation des sauvegardes en modifiant le cluster à l'aide du AWS Management Console ou du AWS CLI. Pour plus d’informations, consultez Modification d'un cluster Amazon DocumentDB.

Copier le chiffrement des instantanés du cluster

Le chiffrement des clusters et des snapshots repose sur une clé de chiffrement KMS. L'ID de clé KMS est l'Amazon Resource Name (ARN), l'identifiant de clé KMS ou l'alias de clé KMS pour la clé de chiffrement KMS.

Les directives et restrictions suivantes s'appliquent :

- Le chiffrement est déduit du cluster lors de la création d'un instantané. Si le cluster est chiffré, le snapshot de ce cluster est chiffré avec la même clé KMS. Si le cluster n'est pas chiffré, le snapshot n'est pas chiffré.
- Si vous copiez un instantané de cluster chiffré depuis votre compte Amazon Web Services, vous pouvez spécifier une valeur `KmsKeyId` pour chiffrer la copie avec une nouvelle clé de chiffrement KMS. Si vous ne spécifiez aucune valeur pour`KmsKeyId`, la copie de l'instantané du cluster est chiffrée avec la même clé KMS que l'instantané du cluster source.
- Si vous copiez un instantané de cluster chiffré partagé depuis un autre compte Amazon Web Services, vous devez spécifier une valeur pour`KmsKeyId`.
- Pour copier un instantané de cluster chiffré vers une autre région Amazon Web Services, définissez `KmsKeyId` l'ID de clé KMS que vous souhaitez utiliser pour chiffrer la copie de l'instantané de cluster dans la région de destination. Les clés de chiffrement KMS sont spécifiques à la région Amazon Web Services dans laquelle elles sont créées, et vous ne pouvez pas utiliser les clés de chiffrement d'une région Amazon Web Services dans une autre région Amazon Web Services.
- Si vous copiez un instantané de cluster non chiffré et que vous spécifiez une valeur pour le `KmsKeyId` paramètre, une erreur est renvoyée.

Comparaison d'instantanés manuels et automatiques

Les principales fonctionnalités des instantanés automatiques et manuels d'Amazon DocumentDB (compatibles avec MongoDB) sont les suivantes.

Les instantanés automatiques Amazon DocumentDB présentent les principales fonctionnalités suivantes :

- Dénomination automatique des instantanés : les noms automatiques des instantanés `yyyy-mm-dd-hh-mm` suivent le modèle`rds:<cluster-name>-yyyy-mm-dd-hh-mm`, en indiquant la date et l'heure de création de l'instantané.
- Créé automatiquement selon un calendrier — Lorsque vous créez ou modifiez un cluster, vous pouvez définir la période de conservation des sauvegardes sur une valeur entière comprise entre 1 et 35 jours. Par défaut, les nouveaux clusters ont une période de rétention des sauvegardes de 1 jour. La période de rétention des sauvegardes définit le nombre de jours pendant lesquels les instantanés automatiques sont conservés avant leur suppression automatique. Vous ne pouvez pas désactiver les sauvegardes automatiques sur les clusters Amazon DocumentDB.

Outre la définition de la période de rétention des sauvegardes, vous définissez également le créneau de sauvegarde, l'heure du jour pendant laquelle les instantanés automatiques sont créés.

- Suppression des instantanés automatiques : les instantanés automatiques sont supprimés lorsque vous supprimez le cluster des instantanés automatiques. Vous ne pouvez pas supprimer manuellement un instantané automatique.
- Incrémentiel : pendant la période de conservation des sauvegardes, les mises à jour de la base de données sont enregistrées afin qu'il existe un enregistrement incrémentiel des modifications.
- Restauration à partir d'un instantané automatique : vous pouvez effectuer une restauration à partir d'un instantané automatique à l'aide du AWS Management Console ou du AWS CLI. Lorsque vous effectuez une restauration à partir d'un instantané à l'aide du AWS CLI, vous devez ajouter des instances séparément une fois le cluster disponible.
- Partage : vous ne pouvez pas partager un instantané de cluster automatisé Amazon DocumentDB. Pour contourner le problème, vous pouvez créer un instantané manuel en copiant l'instantané automatisé, puis en partageant cette copie. Pour de plus amples informations sur la copie d'un instantané, veuillez consulter Copier des instantanés du cluster Amazon DocumentDB. Pour de plus amples informations sur la restauration d'un cluster à partir d’un instantané, veuillez consulter Restauration d'un cluster à partir d'un instantané.
- Vous pouvez effectuer une restauration à tout moment pendant la période de conservation des sauvegardes : les mises à jour de base de données étant enregistrées de manière incrémentielle, vous pouvez restaurer votre cluster à tout moment pendant la période de conservation des sauvegardes.

  Lorsque vous effectuez une restauration à partir d'un instantané automatique ou d'une point-in-time restauration à l'aide du AWS CLI, vous devez ajouter des instances séparément une fois le cluster disponible.

Les instantanés manuels Amazon DocumentDB présentent les principales fonctionnalités suivantes :

- Créé à la demande — Les instantanés manuels Amazon DocumentDB sont créés à la demande à l'aide de la console de gestion Amazon DocumentDB ou. AWS CLI
- Suppression d'un instantané manuel : un instantané manuel est supprimé uniquement lorsque vous le supprimez explicitement à l'aide de la console Amazon DocumentDB ou. AWS CLI Un instantané manuel n'est pas supprimé lorsque vous supprimez son cluster.
- Sauvegardes complètes : lorsqu'un instantané est pris manuellement, une sauvegarde complète des données de votre cluster est créée et stockée.

- Dénomination manuelle des instantanés : vous spécifiez le nom de l'instantané manuel. Amazon DocumentDB n'ajoute pas de `datetime` tampon au nom. Vous devez donc ajouter ces informations si vous souhaitez qu'elles soient incluses dans le nom.
- Restauration à partir d'un instantané manuel : vous pouvez effectuer une restauration à partir d'un instantané manuel à l'aide de la console ou du AWS CLI. Lorsque vous effectuez une restauration à partir d'un instantané à l'aide du AWS CLI, vous devez ajouter des instances séparément une fois le cluster disponible.
- Quotas de service — Vous êtes limité à un maximum de 100 instantanés manuels par. Région AWS
- Partage : vous pouvez partager des instantanés de cluster manuels, qui peuvent être copiés par des personnes autorisées Comptes AWS. Vous pouvez partager des instantanés manuels chiffrés ou non chiffrés. Pour de plus amples informations sur la copie d'un instantané, veuillez consulter Copier des instantanés du cluster Amazon DocumentDB.
- Vous restaurez à la date à laquelle l'instantané manuel a été pris : lorsque vous restaurez à partir d'un instantané manuel, vous restaurez à la date à laquelle l'instantané manuel a été pris.

  Lorsque vous effectuez une restauration à partir d'un instantané à l'aide du AWS CLI, vous devez ajouter des instances séparément une fois le cluster disponible.

Création d'un instantané manuel d'un cluster

Vous pouvez créer un instantané manuel à l'aide du AWS Management Console ou AWS CLI. Le temps nécessaire à la création d'un instantané varie en fonction de la taille de vos bases de données. Lorsque vous créez un instantané, vous devez procéder comme suit :

1. Identifier le cluster à sauvegarder.
2. Donner un nom à votre instantané. Cela vous permet d'exécuter ensuite une restauration à partir de ce dernier.

Using the AWS Management Console

Pour créer un instantané manuel à l'aide de AWS Management Console, vous pouvez suivre l'une des méthodes ci-dessous.

1. Méthode 1 :

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Snapshots (Instantanés).

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Sur la page Instantanés, choisissez Créer.
4. Sur la page Créer un instantané de cluster :
   a. Identifiant du cluster : dans la liste déroulante des clusters, choisissez le cluster dont vous souhaitez créer un instantané.
   b. Identifiant de capture d'écran : entrez un nom pour votre instantané.

      Contraintes d'attribution de nom relatives à un instantané :

      - La longueur est de [1 à 255] lettres, chiffres ou traits d'union.
      - Le premier caractère doit être une lettre.
      - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
      - Doit être unique pour tous les clusters (sur Amazon RDS, Amazon Neptune et Amazon DocumentDB) AWS par compte et par région.

   c. Choisissez Créer.

2. Méthode 2 :

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu

(≡)
dans le coin supérieur gauche de la page.

3. Sur la page Clusters, choisissez le bouton sur la gauche du cluster dont vous voulez créer un instantané.
4. Dans le menu Actions, choisissez Take snapshot (Prendre un instantané).
5. Sur la page Créer un instantané de cluster :
   a. Identifiant de capture d'écran : entrez un nom pour votre instantané.

      Contraintes d'attribution de nom relatives à un instantané :

      - La longueur est de [1 à 63] lettres, chiffres ou traits d'union.
      - Le premier caractère doit être une lettre.
      - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
      - Doit être unique pour tous les clusters (sur Amazon RDS, Amazon Neptune et Amazon DocumentDB) AWS par compte et par région.

   b. Choisissez Créer.

Using the AWS CLI

Pour créer un instantané de cluster à l'aide de AWS CLI, utilisez l'`create-db-cluster-snapshot`opération avec les paramètres suivants.

Paramètres

- **`--db-cluster-identifier`** — Obligatoire. Le nom du cluster dont vous prenez un instantané. Ce cluster doit exister et être disponible.
- **`--db-cluster-snapshot-identifier`** — Obligatoire. Le nom de l'instantané manuel en cours de création.

L'exemple suivant crée un instantané nommé `sample-cluster-snapshot` pour un cluster nommé `sample-cluster`.

Pour Linux, macOS ou Unix :

```
aws docdb create-db-cluster-snapshot \
    --db-cluster-identifier sample-cluster \
```

```
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Pour Windows :

```
aws docdb create-db-cluster-snapshot ^
    --db-cluster-identifier sample-cluster ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1c"
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
        "DBClusterIdentifier": "sample-cluster",
        "SnapshotCreateTime": "2020-04-24T04:59:08.475Z",
        "Engine": "docdb",
        "Status": "creating",
        "Port": 0,
        "VpcId": "vpc-abc0123",
        "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
        "MasterUsername": "master-user",
        "EngineVersion": "4.0.0",
        "SnapshotType": "manual",
        "PercentProgress": 0,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:<accountID>:cluster-
snapshot:sample-cluster-snapshot"
    }
}
```

Copier des instantanés du cluster Amazon DocumentDB

Dans Amazon DocumentDB, vous pouvez copier des instantanés manuels et automatiques au sein du même compte Région AWS ou vers un autre compte Région AWS au sein du même

compte. Vous pouvez également partager des instantanés appartenant Comptes AWS à d'autres utilisateurs. Région AWS Toutefois, vous ne pouvez pas copier un instantané de cluster d'un bout à l'autre Régions AWS et Compte AWS en une seule étape. Ces actions doivent être effectuées individuellement.

Au lieu de copier, vous pouvez également partager des instantanés manuels avec d'autres Comptes AWS personnes. Pour plus d'informations, consultez Partage des instantanés du cluster Amazon DocumentDB.

> **Note**
>
> Amazon DocumentDB vous facture en fonction de la quantité de données de sauvegarde et de capture instantanée que vous conservez et de la durée pendant laquelle vous les conservez. Pour plus d'informations sur le stockage associé aux sauvegardes et aux instantanés Amazon DocumentDB, consultez. Présentation de l'utilisation du stockage de sauvegarde Pour obtenir des informations sur la tarification du stockage Amazon DocumentDB, consultez la section Tarification d'Amazon DocumentDB.

Rubriques

Copie d'instantanés partagés

Vous pouvez copier des instantanés que d'autres Comptes AWS personnes vous ont partagés. Si vous copiez un instantané chiffré qui a été partagé depuis un autre Compte AWS, vous devez avoir accès à la clé de AWS KMS chiffrement utilisée pour chiffrer le cliché.

Vous ne pouvez copier qu'un instantané partagé dans le même Région AWS document, qu'il soit chiffré ou non. Pour plus d'informations, consultez Chiffrement.

Copier des instantanés Régions AWS

Lorsque vous copiez un instantané vers un cliché différent de Région AWS celui de la source Région AWS, chaque copie est un instantané complet. Une copie instantanée complète contient toutes les données et métadonnées requises pour restaurer le cluster Amazon DocumentDB.

En fonction du type Régions AWS concerné et de la quantité de données à copier, la réalisation d'une copie instantanée entre régions peut prendre des heures. Dans certains cas, il peut y avoir un grand nombre de demandes de copie instantanée entre régions provenant d'une source Région AWS donnée. Dans ces cas, Amazon DocumentDB peut placer les nouvelles demandes de copie entre régions provenant de cette source Région AWS dans une file d'attente jusqu'à ce que certaines copies en cours soient terminées. Aucune information d'avancement n'est affichée sur les demandes de copie quand elles sont en file d'attente. Les informations d'avancement sont affichées lorsque la copie commence.

Limites

Vous trouverez ci-dessous certaines limites qui s'appliquent lorsque vous copiez des instantanés :

- Si vous supprimez un instantané source avant que l'instantané cible soit disponible, la copie d'instantané peut échouer. Vérifiez que l'instantané cible a le statut `AVAILABLE` avant de supprimer un instantané source.
- Vous pouvez avoir jusqu'à cinq demandes de copie d'instantanés en cours vers une même région de destination par compte.
- Selon les régions impliquées et le volume de données à copier, la copie d'un instantané entre régions peut prendre plusieurs heures. Pour plus d’informations, consultez Copier des instantanés Régions AWS.

Chiffrement

Vous pouvez copier un instantané qui a été chiffré à l'aide d'une clé de chiffrement AWS KMS . Si vous copiez un instantané chiffré, la copie de l'instantané doit également être chiffrée. Si vous copiez un instantané chiffré dans le même instantané Région AWS, vous pouvez chiffrer la copie avec la même clé de AWS KMS chiffrement que l'instantané d'origine, ou vous pouvez spécifier une clé de AWS KMS chiffrement différente. Si vous copiez un instantané chiffré d'une région à l'autre, vous ne pouvez pas utiliser la même clé de AWS KMS chiffrement pour la copie que pour l'instantané source,

car AWS KMS les clés sont spécifiques à chaque région. Vous devez plutôt spécifier une AWS KMS clé valide dans le Région AWS n de destination.

L'instantané source reste chiffré pendant tout le processus de copie. Pour plus d'informations, consultez Protection des données dans Amazon DocumentDB.

> **Note**
>
> Pour les instantanés de cluster Amazon DocumentDB, vous ne pouvez pas chiffrer un instantané de cluster non chiffré lorsque vous le copiez.

Considérations relatives au groupe de paramètres

Lorsque vous copiez un instantané entre plusieurs régions, la copie n'inclut pas le groupe de paramètres utilisé par le cluster Amazon DocumentDB d'origine. Lorsque vous restaurez un instantané pour créer un nouveau cluster, ce cluster obtient le groupe de paramètres par défaut pour le cluster dans Région AWS lequel il a été créé. Pour attribuer au nouveau cluster les mêmes paramètres que l'original, vous devez effectuer les opérations suivantes :

1. Dans la destination Région AWS, créez un groupe de paramètres de cluster Amazon DocumentDB avec les mêmes paramètres que le cluster d'origine. S'il en existe déjà un dans le nouveau Région AWS, vous pouvez l'utiliser.
2. Après avoir restauré l'instantané dans la destination Région AWS, modifiez le nouveau cluster Amazon DocumentDB et ajoutez le groupe de paramètres nouveau ou existant de l'étape précédente. Pour plus d'informations, consultez Modification d'un cluster Amazon DocumentDB.

Copie d'un instantané de cluster

Vous pouvez copier un cluster Amazon DocumentDB à l'aide du AWS Management Console ou AWS CLI, comme suit.

Using the AWS Management Console

Pour créer une copie d'un instantané de cluster à l'aide du AWS Management Console, procédez comme suit. Cette procédure fonctionne pour copier des instantanés de cluster chiffrés ou non chiffrés, dans la même région Région AWS ou entre plusieurs régions.

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le volet de navigation, choisissez Snapshots, puis cliquez sur le bouton situé à gauche de l'instantané que vous souhaitez copier.

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans le menu Actions, choisissez Copy (Copier).
4. Dans la page Créer une copie d'un instantané du cluster qui s'affiche, complétez la section Paramètres.

   a. Région de destination — Facultatif. Pour copier le cliché du cluster vers un autre Région AWS, choisissez-le Région AWS pour Région de destination.

   b. Identifiant du nouvel instantané : entrez le nom du nouveau cliché.

      Contraintes d'attribution de nom relatives à un instantané cible :

      - Ne peut pas être le nom d'un instantané existant.
      - La longueur est de [1 à 63] lettres, chiffres ou traits d'union.
      - Le premier caractère doit être une lettre.
      - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
      - Doit être unique pour tous les clusters d'Amazon RDS, Neptune et Amazon Compte AWS DocumentDB par région.

   c. Copier les balises : pour copier les balises présentes sur votre instantané source vers votre copie instantanée, choisissez Copier les balises.

5. Complétez la ncryption-at-rest section E.

   a. Chiffrement au repos : si votre instantané n'est pas chiffré, vous ne pouvez pas accéder à ces options car vous ne pouvez pas créer de copie chiffrée à partir d'un instantané non chiffré. Si votre instantané est chiffré, vous pouvez modifier le paramètre AWS KMS key utilisé pendant le chiffrement au repos.

Pour plus d'informations sur le chiffrement des copies instantanées, consultezCopier le chiffrement des instantanés du cluster.

Pour plus d'informations sur le chiffrement au repos, veuillez consulter Chiffrement des données Amazon DocumentDB au repos.

b. AWS KMS Clé : dans la liste déroulante, sélectionnez l'une des options suivantes :

- (par défaut) aws/rds — Le numéro de compte et l'ID de AWS KMS clé sont répertoriés après cette option.
- < some-key-name > — Si vous avez créé une clé, elle est répertoriée et vous pouvez la choisir.
- Entrez un ARN de clé — Dans le champ ARN, entrez le nom de ressource Amazon (ARN) pour votre AWS KMS clé. Le format de l'ARN est : `arn:aws:kms:<region>:<accountID>:key/<key-id>` .

6. Pour créer une copie de l'instantané sélectionné, choisissez Copy snapshot (Copier un instantané). Vous pouvez également choisir Annuler pour ne pas faire de copie de l'instantané.

Using the AWS CLI

Pour créer une copie d'un instantané de cluster non chiffré à l'aide de AWS CLI, utilisez l'`copy-db-cluster-snapshot`opération avec les paramètres suivants. Si vous copiez le cliché vers un autre Région AWS, exécutez la commande dans laquelle le cliché sera copié. Région AWS

- **`--source-db-cluster-snapshot-identifier`** — Obligatoire. L'identifiant de l'instantané du cluster à copier. L'instantané du cluster doit exister et se trouver à l'état disponible. Si vous copiez l'instantané vers un autre Région AWS, cet identifiant doit être au format ARN de la source Région AWS. Ce paramètre n'est pas sensible à la casse.
- **`--target-db-cluster-snapshot-identifier`** — Obligatoire. L'identifiant du nouvel instantané du cluster à créer à partir de l'instantané du cluster source. Ce paramètre n'est pas sensible à la casse.

  Contraintes d'attribution de nom relatives à un instantané cible :

  - Ne peut pas être le nom d'un instantané existant.
  - La longueur est de [1 à 63] lettres, chiffres ou traits d'union.
  - Le premier caractère doit être une lettre.

  - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
  - Doit être unique pour tous les clusters d'Amazon RDS, Neptune et Amazon Compte AWS DocumentDB par région.
- **`--source-region`**— Si vous copiez le cliché vers un autre Région AWS, spécifiez à partir Région AWS duquel le cliché du cluster chiffré sera copié.

  Si vous copiez l'instantané vers un autre Région AWS et que vous ne le spécifiez pas`--source-region`, vous devez spécifier l'`pre-signed-url`option à la place. La `pre-signed-url` valeur doit être une URL contenant une demande signée Signature version 4 pour que l'`CopyDBClusterSnapshot`action soit appelée dans la source à partir de Région AWS laquelle le cliché du cluster est copié. Pour en savoir plus sur le`pre-signed-url`, consultez CopyDB. ClusterSnapshot
- **`--kms-key-id`**— Identifiant de clé KMS correspondant à la clé à utiliser pour chiffrer la copie de l'instantané du cluster.

  Si vous copiez un instantané de cluster chiffré vers un autre Région AWS, ce paramètre est obligatoire. Vous devez spécifier une clé KMS pour la destination Région AWS.

  Si vous copiez un instantané de cluster chiffré dans celui-ci Région AWS, le paramètre AWS KMS clé est facultatif. La copie de l'instantané du cluster est chiffrée avec la même AWS KMS clé que l'instantané du cluster source. Si vous souhaitez spécifier une nouvelle clé de AWS KMS chiffrement à utiliser pour chiffrer la copie, vous pouvez le faire à l'aide de ce paramètre.
- **`--copy-tags`**— Facultatif. Les balises et les valeurs à copier.

Pour annuler une opération de copie une fois qu'elle est en cours, vous pouvez supprimer le cliché de cluster cible identifié par `--target-db-cluster-snapshot-identifier` ou `TargetDBClusterSnapshotIdentifier` pendant que cet instantané de cluster est en état de copie.

Example

Exemple 1 : Copier un instantané non chiffré dans la même région

L' AWS CLI exemple suivant crée une copie de `sample-cluster-snapshot` named `sample-cluster-snapshot-copy` in Région AWS identique à l'instantané source. Lorsque la copie est réalisée, toutes les balises de l'instantané d'origine sont copiées dans la copie de l'instantané.

Pour Linux, macOS ou Unix :

```
aws docdb copy-db-cluster-snapshot \
    --source-db-cluster-snapshot-identifier sample-cluster-snapshot \
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy \
    --copy-tags
```

Pour Windows :

```
aws docdb copy-db-cluster-snapshot ^
    --source-db-cluster-snapshot-identifier sample-cluster-snapshot ^
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy ^
    --copy-tags
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1c"
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot-copy",
        "DBClusterIdentifier": "sample-cluster",
        "SnapshotCreateTime": "2020-03-27T08:40:24.805Z",
        "Engine": "docdb",
        "Status": "copying",
        "Port": 0,
        "VpcId": "vpc-abcd0123",
        "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
        "MasterUsername": "master-user",
        "EngineVersion": "4.0.0",
        "SnapshotType": "manual",
        "PercentProgress": 0,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:111122223333:key/sample-key-id",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot-copy",
        "SourceDBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot"
    }
}
```

Example

Exemple 2 : Copier un instantané non chiffré sur Régions AWS

L' AWS CLI exemple suivant crée une copie de`sample-cluster-snapshot`, qui possède l'ARN`arn:aws:rds:us-east-1:123456789012:cluster-snapshot:sample-cluster-snapshot`. Cette copie est nommée named `sample-cluster-snapshot-copy` et se trouve dans le Région AWS fichier dans lequel la commande est exécutée.

Pour Linux, macOS ou Unix :

```
aws docdb copy-db-cluster-snapshot \
    --source-db-cluster-snapshot-identifier arn:aws:rds:us-
east-1:123456789012:cluster-snapshot:sample-cluster-snapshot \
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy
```

Pour Windows :

```
aws docdb copy-db-cluster-snapshot ^
    --source-db-cluster-snapshot-identifier arn:aws:rds:us-
east-1:123456789012:cluster-snapshot:sample-cluster-snapshot ^
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1c"
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot-copy",
        "DBClusterIdentifier": "sample-cluster",
        "SnapshotCreateTime": "2020-04-29T16:45:51.239Z",
        "Engine": "docdb",
        "AllocatedStorage": 0,
        "Status": "copying",
        "Port": 0,
        "VpcId": "vpc-abc0123",
        "ClusterCreateTime": "2020-04-28T16:43:00.294Z",
        "MasterUsername": "master-user",
```

```
        "EngineVersion": "4.0.0",
        "LicenseModel": "docdb",
        "SnapshotType": "manual",
        "PercentProgress": 0,
        "StorageEncrypted": false,
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot-copy",
        "SourceDBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot",
    }
}
```

Example

Exemple 3 : copier un instantané chiffré sur Régions AWS

L' AWS CLI exemple suivant crée une copie `sample-cluster-snapshot` de la région us-west-2 vers la région us-east-1. Cette commande est appelée dans la région us-east-1.

Pour Linux, macOS ou Unix :

```
aws docdb copy-db-cluster-snapshot \
    --source-db-cluster-snapshot-identifier arn:aws:rds:us-
west-2:123456789012:cluster-snapshot:sample-cluster-snapshot \
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy \
    --source-region us-west-2 \
    --kms-key-id sample-us-east-1-key
```

Pour Windows :

```
aws docdb copy-db-cluster-snapshot ^
    --source-db-cluster-snapshot-identifier arn:aws:rds:us-
west-2:123456789012:cluster-snapshot:sample-cluster-snapshot ^
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy ^
    --source-region us-west-2 ^
    --kms-key-id sample-us-east-1-key
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [],
```

```
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot-copy",
        "DBClusterIdentifier": "ayhu-xrsc-test-ap-southeast-1-small-cluster-kms",
        "SnapshotCreateTime": "2020-04-29T16:45:53.159Z",
        "Engine": "docdb",
        "AllocatedStorage": 0,
        "Status": "copying",
        "Port": 0,
        "ClusterCreateTime": "2020-04-28T16:43:07.129Z",
        "MasterUsername": "chimera",
        "EngineVersion": "4.0.0",
        "LicenseModel": "docdb",
        "SnapshotType": "manual",
        "PercentProgress": 0,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:111122223333:key/sample-key-id",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot-copy",
        "SourceDBClusterSnapshotArn": "arn:aws:rds:us-west-2:111122223333:cluster-
snapshot:sample-cluster-snapshot",
    }
}
```

> **Note**
>
> Pour plus d'informations sur le chiffrement des copies instantanées, consultezCopier le chiffrement des instantanés du cluster.
> Pour plus d'informations sur le chiffrement au repos, veuillez consulter Chiffrement des données Amazon DocumentDB au repos.

Partage des instantanés du cluster Amazon DocumentDB

Dans Amazon DocumentDB, vous pouvez partager des instantanés de cluster manuels, qui peuvent être copiés par des personnes autorisées. Comptes AWS Vous pouvez partager des instantanés manuels chiffrés ou non chiffrés. Lors du partage d'un instantané non chiffré, les personnes autorisées Comptes AWS peuvent restaurer le cluster directement à partir de l'instantané au lieu d'en faire une copie et de le restaurer à partir de celui-ci. Cependant, vous ne pouvez pas restaurer un cluster à partir d'un instantané qui est à la fois partagé et chiffré. Par contre, vous pouvez créer une copie du cluster et restaurer le cluster à partir de cette copie. Pour de plus amples informations sur la copie d'un instantané, veuillez consulter Copier des instantanés du cluster Amazon DocumentDB.

> **Note**
>
> Vous ne pouvez pas partager un instantané de cluster automatisé Amazon DocumentDB. Pour contourner le problème, vous pouvez créer un instantané manuel en copiant l'instantané automatisé, puis en partageant cette copie. Pour de plus amples informations sur la copie d'un instantané, veuillez consulter Copier des instantanés du cluster Amazon DocumentDB. Pour de plus amples informations sur la restauration d'un cluster à partir d'un instantané, veuillez consulter Restauration d'un cluster à partir d'un instantané.

Vous pouvez partager un instantané manuel avec un maximum de 20 autres personnes Comptes AWS. Vous pouvez également partager un instantané manuel non chiffré marqué comme public ; il est ainsi accessible à tous les comptes . Lors du partage d'un instantané marqué comme public, assurez-vous de n'inclure aucune information privée dans vos instantanés publics.

Lorsque vous partagez des instantanés manuels avec d'autres Comptes AWS utilisateurs et que vous restaurez un cluster à partir d'un instantané partagé à l' AWS CLI aide de l'API Amazon DocumentDB, vous devez spécifier le nom de ressource Amazon (ARN) du cliché partagé comme identifiant d'instantané.

Partage d'un instantané chiffré

Les restrictions suivantes s'appliquent au partage d'instantanés chiffrés :

- Vous ne pouvez pas partager des instantanés chiffrés marqués comme publics.
- Vous ne pouvez pas partager un instantané chiffré à l'aide de la clé de AWS KMS chiffrement par défaut du compte qui a partagé l'instantané.

Suivez les étapes ci-après pour partager des instantanés chiffrés.

1. Partagez la clé de chiffrement AWS Key Management Service (AWS KMS) qui a été utilisée pour chiffrer l'instantané avec tous les comptes auxquels vous souhaitez accéder à l'instantané.

   Vous pouvez partager des clés de AWS KMS chiffrement avec d'autres AWS comptes en ajoutant les autres comptes à la politique des AWS KMS clés. Pour plus de détails sur la mise à jour d'une politique clé, consultez la section Utilisation des politiques clés dans AWS KMS dans le Guide du AWS Key Management Service développeur. Pour obtenir un exemple de

création d'une stratégie de clé, consultez Création d'une stratégie IAM pour permettre la copie d'un instantané chiffré plus loin dans cette rubrique.

2. Utilisez le AWS CLI, comme indiqué ci-dessous, pour partager l'instantané chiffré avec les autres comptes.

Autoriser l'accès à une clé AWS KMS de chiffrement

Pour Compte AWS qu'un autre puisse copier un instantané chiffré partagé depuis votre compte, le compte avec lequel vous partagez votre instantané doit avoir accès à la AWS KMS clé qui a chiffré l'instantané. Pour autoriser un autre compte à accéder à une AWS KMS clé, mettez à jour la politique de AWS KMS clé pour la clé avec l'ARN du compte que vous partagez en tant que principal dans la politique de AWS KMS clé. Autorisez ensuite l'action `kms:CreateGrant`.

Une fois que vous avez accordé à un compte l'accès à votre clé de AWS KMS chiffrement, pour copier votre instantané chiffré, ce compte doit créer un utilisateur AWS Identity and Access Management (IAM) s'il n'en possède pas déjà un. En outre, ce compte doit également associer une politique IAM à cet utilisateur IAM lui permettant de copier un instantané chiffré à l'aide de votre AWS KMS clé. Le compte doit être un utilisateur IAM et ne peut pas être une Compte AWS identité root en raison de restrictions de AWS KMS sécurité.

Dans l'exemple de politique de clé suivant, l'utilisateur 123451234512 est le propriétaire de la clé de chiffrement. AWS KMS L'utilisateur 123456789012 est le compte avec lequel la clé est partagée. Cette politique clé mise à jour permet au compte d'accéder à la AWS KMS clé. Pour ce faire, il inclut l'ARN de l' Compte AWS identité racine de l'utilisateur 123456789012 en tant que principal de la politique et en autorisant l'action. `kms:CreateGrant`

```
{
    "Id": "key-policy-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {"AWS": [
                "arn:aws:iam::123451234512:user/KeyUser",
                "arn:aws:iam::123456789012:root"
            ]},
            "Action": [
                "kms:CreateGrant",
                "kms:Encrypt",
```

```
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "*"},
            {
            "Sid": "Allow attachment of persistent resources",
            "Effect": "Allow",
            "Principal": {"AWS": [
                "arn:aws:iam::123451234512:user/KeyUser",
                "arn:aws:iam::123456789012:root"
            ]},
            "Action": [
                "kms:CreateGrant",
                "kms:ListGrants",
                "kms:RevokeGrant"
            ],
            "Resource": "*",
            "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
        }
    ]
}
```

Création d'une stratégie IAM pour permettre la copie d'un instantané chiffré

Lorsque l'utilisateur externe Compte AWS a accès à votre AWS KMS clé, le propriétaire de ce compte peut créer une politique permettant à un utilisateur IAM créé pour le compte de copier un instantané chiffré avec cette AWS KMS clé.

L'exemple suivant montre une politique qui peut être attachée à un utilisateur IAM pour Compte AWS 123456789012. La politique permet à l'utilisateur IAM de copier un instantané partagé à partir du compte 123451234512 qui a été chiffré avec la clé AWS KMS dans `c989c1dd-a3f2-4a5d-8d96-e793d082ab26` la région us-west-2.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowUseOfTheKey",
            "Effect": "Allow",
            "Action": [
```

```
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey",
                "kms:CreateGrant",
                "kms:RetireGrant"
            ],
            "Resource": ["arn:aws:kms:us-west-2:123451234512:key/c989c1dd-
a3f2-4a5d-8d96-e793d082ab26"]
        },
        {
            "Sid": "AllowAttachmentOfPersistentResources",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:ListGrants",
                "kms:RevokeGrant"
            ],
            "Resource": ["arn:aws:kms:us-west-2:123451234512:key/c989c1dd-
a3f2-4a5d-8d96-e793d082ab26"],
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
            }
        }
    ]
}
```

Pour plus de détails sur la mise à jour d'une politique clé, consultez la section Utilisation des politiques clés AWS KMS dans le guide du AWS Key Management Service développeur.

Partage d'un instantané

Pour partager un instantané, utilisez l'opération Amazon DocumentDB. `modify-db-snapshot-attribute` Utilisez le `--values-to-add` paramètre pour ajouter une liste des identifiants autorisés à restaurer l'instantané manuel. Comptes AWS

L'exemple suivant permet à deux Compte AWS identifiants, 123451234512 et 123456789012, de restaurer le cliché nommé. `manual-snapshot1` Il supprime également la valeur d'attribut `all` pour marquer l'instantané comme étant privé.

Pour Linux, macOS ou Unix :

```
aws docdb modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier sample-cluster-snapshot \
    --attribute-name restore \
    --values-to-add '["123451234512","123456789012"]'
```

Pour Windows :

```
aws docdb modify-db-cluster-snapshot-attribute ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot ^
    --attribute-name restore ^
    --values-to-add '["123451234512","123456789012"]'
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123451234512",
                    "123456789012"
                ]
            }
        ]
    }
}
```

Pour supprimer un Compte AWS identifiant de la liste, utilisez le `--values-to-remove` paramètre. L'exemple suivant empêche l' Compte AWS ID 123456789012 de restaurer le snapshot.

Pour Linux, macOS ou Unix :

```
aws docdb modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier sample-cluster-snapshot \
    --attribute-name restore \
    --values-to-remove '["123456789012"]'
```

Pour Windows :

```
aws docdb modify-db-cluster-snapshot-attribute ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot ^
    --attribute-name restore ^
    --values-to-remove '["123456789012"]'
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123451234512"
                ]
            }
        ]
    }
}
```

Restauration d'un cluster à partir d'un instantané

Amazon DocumentDB (compatible avec MongoDB) crée un instantané de cluster de votre volume de stockage. Vous pouvez créer un nouveau cluster en le restaurant à partir de l'instantané de cluster. Lorsque vous restaurez le cluster, vous indiquez le nom de l'instantané de cluster à partir duquel la restauration doit être effectuée, et le nom du nouveau cluster créé par la restauration. Vous ne pouvez pas restaurer un instantané vers un cluster existant, car un nouveau cluster est créé lors de la restauration.

Lorsque vous restaurez un cluster à partir d'un instantané :

- Cette action restaure uniquement le cluster, mais pas les instances de ce cluster. Vous devez invoquer l'action `create-db-instance` pour créer des instances pour le cluster restauré, en spécifiant l'identifiant du cluster restauré dans `--db-cluster-identifier`. Vous pouvez créer des instances uniquement une fois que le cluster est disponible.
- Vous ne pouvez pas restaurer un instantané chiffré dans un cluster non chiffré. Toutefois, vous pouvez restaurer un instantané non chiffré sur un cluster chiffré en spécifiant la AWS KMS clé.

- Pour restaurer un cluster à partir d'un instantané chiffré, vous devez avoir accès à la AWS KMS clé.

> **Note**
>
> Vous ne pouvez pas restaurer un cluster 3.6 vers un cluster 4.0, mais vous pouvez migrer d'une version de cluster à une autre. Pour plus d'informations, consultez la section concernant Migration vers Amazon DocumentDB.

Using the AWS Management Console

La procédure suivante explique comment restaurer un cluster Amazon DocumentDB à partir d'un instantané de cluster à l'aide de la console de gestion Amazon DocumentDB.

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le volet de navigation, choisissez Instantanés, puis sélectionnez le bouton sur la gauche l'instantané que vous voulez utiliser pour restaurer un cluster.

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans le menu Actions , choisissez Restaurer.
4. Sur la page Restore snapshot (Restaurer l'instantané), renseignez la Configuration.

   a. Identifiant du cluster : nom du nouveau cluster. Vous pouvez accepter le nom fourni par Amazon DocumentDB ou saisir le nom de votre choix. Le nom Amazon DocumentDBSupplied est au format `docdb-` plus un horodatage UTC ; par exemple,. `docdb-yyyy-mm-dd-hh-mm-ss`

   b. Classe d'instance : classe d'instance du nouveau cluster. Vous pouvez accepter la classe d'instance par défaut ou choisir une classe d'instance dans la liste déroulante.

c. Nombre d'instances : nombre d'instances que vous souhaitez créer avec ce cluster. Vous pouvez accepter la valeur par défaut de 3 instances (1 nœud principal en lecture/écriture et 2 réplicas en lecture seule) ou choisir dans la liste déroulante le nombre d'instances que vous voulez créer avec ce cluster.

5. Pour la configuration du stockage en cluster, choisissez une option de stockage.

> **Note**
>
> La configuration de stockage optimisée pour les E/S d'Amazon DocumentDB n'est disponible que sur la version 5.0 du moteur Amazon DocumentDB.

6. Si la configuration du cluster vous convient, choisissez Restore cluster (Restaurer le cluster) et patientez pendant la restauration du cluster.
7. Si vous préférez modifier certaines configurations, par exemple en spécifiant un VPC ou un groupe de sécurité Amazon autre que celui par défaut, choisissez Afficher les paramètres avancés en bas à gauche de la page, puis passez aux étapes suivantes.

   a. Complétez la section Network settings (Paramètres de réseau).

      - Virtual Private Cloud (VPC) : acceptez le VPC actuel ou choisissez-en un dans la liste déroulante.
      - Groupe de sous-réseaux : acceptez le groupe de `default` sous-réseaux ou choisissez-en un dans la liste déroulante.
      - Groupes de sécurité VPC : acceptez le groupe de `default (VPC)` sécurité ou choisissez-en un dans la liste.

   b. Complétez la section Cluster options (Options du cluster).

      - Port de base de données : acceptez le port par défaut ou utilisez la flèche vers le haut ou vers le bas pour définir le port que vous souhaitez utiliser pour les connexions aux applications. `27017`

   c. Complétez la section Encryption (Chiffrement).

      - Chiffrement au repos : si votre instantané est chiffré, vous ne pouvez pas accéder à ces options. Si elle ne le sont pas, vous pouvez choisir l'une des actions suivantes :
        - Pour chiffrer toutes les données de votre cluster, sélectionnez Activer encryption-at-rest. Si vous choisissez cette option, vous devez désigner une clé KMS.

- Pour ne pas chiffrer les données de votre cluster, choisissez Désactiver encryption-at-rest. Si vous choisissez cette option, vous avez terminé la section de chiffrement.

- AWS KMS Clé — Choisissez l'une des options suivantes dans la liste déroulante :
  - (par défaut) aws/rds — Le numéro de compte et l'ID de AWS KMS clé sont répertoriés après cette option.
  - Clé gérée par le client : cette option n'est disponible que si vous avez créé une clé de chiffrement IAM dans la console AWS Identity and Access Management (IAM). Vous pouvez choisir la clé pour chiffrer votre cluster.
  - Entrez un ARN de clé — Dans le champ ARN, entrez le nom de ressource Amazon (ARN) pour votre AWS KMS clé. Le format de l'ARN est : `arn:aws:kms:<region>:<accountID>:key/<key-id>`.

d. Complétez la section Log exports (Exportations de journaux).

- Sélectionnez les types de journaux sur lesquels publier CloudWatch — Choisissez l'un des types suivants :
  - Activé : permet à votre cluster d'exporter la journalisation DDL vers Amazon CloudWatch Logs.
  - Désactivé : empêche votre cluster d'exporter les journaux DDL vers Amazon CloudWatch Logs. Disabled (Désactivé) est la valeur par défaut.
- Rôle IAM : dans la liste, choisissez RDS Service Linked Role.

e. Complétez la section Tags (Balises).

- Ajouter une étiquette — Dans le champ Clé, entrez le nom de la balise pour votre cluster. Dans la zone Value (Valeur), entrez la valeur de la balise, si vous le souhaitez. Les balises sont utilisées avec les politiques AWS Identity and Access Management (IAM) pour gérer l'accès aux ressources Amazon DocumentDB et pour contrôler les actions qui peuvent être appliquées aux ressources.

f. Complétez la section Deletion protection (Protection contre la suppression) .

- Activer la protection contre la suppression : protège le cluster contre toute suppression accidentelle. Lorsque cette option est activée, vous ne pouvez pas supprimer le cluster.

8. Choisissez Restaurer un cluster.

Using the AWS CLI

Pour restaurer un cluster à partir d'un instantané à l'aide de AWS CLI, utilisez l'`restore-db-cluster-from-snapshot`opération avec les paramètres suivants. Pour plus d'informations, consultez RestoreDBClusterFromSnapshot.

- **`--db-cluster-identifier`** — Obligatoire. Le nom du cluster qui est créé par l'opération. Un cluster portant ce nom ne peut pas exister avant cette opération.

  Contraintes d'attribution de nom relatives à un cluster :

  - La longueur est de [1 à 63] lettres, chiffres ou traits d'union.
  - Le premier caractère doit être une lettre.
  - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
  - Doit être unique pour tous les clusters d'Amazon RDS, Neptune et Amazon Compte AWS DocumentDB par région.
- **`--snapshot-identifier`** — Obligatoire. Le nom de l'instantané utilisé pour la restauration à partir de ce dernier. Un instantané de ce nom doit exister et se trouver à l'état disponible.
- **`--engine`** — Obligatoire. Doit indiquer `docdb`.
- **`--storage-type standard | iopt1`**— Facultatif. Par défaut: `standard`.
- **`--kms-key-id`**— Facultatif. L'ARN de l'identifiant de AWS KMS clé à utiliser lors de la restauration d'un instantané chiffré ou du chiffrement d'un cluster lors de la restauration à partir d'un instantané non chiffré. La fourniture de l'ID de AWS KMS clé entraîne le chiffrement du cluster restauré avec la AWS KMS clé, que l'instantané ait été chiffré ou non.

  Le format de `--kms-key-id` est `arn:aws:kms:<region>:<accountID>:key/<key-id>`. Si vous ne spécifiez pas de valeur pour le paramètre `--kms-key-id` :

  - Si le snapshot `--snapshot-identifier` est chiffré, le cluster restauré est chiffré à l'aide de la même AWS KMS clé que celle utilisée pour chiffrer le snapshot.
  - Si l'instantané spécifié dans `--snapshot-identifier` n'est pas chiffré, le cluster restauré n'est pas chiffré.

Pour Linux, macOS ou Unix :

```
aws docdb restore-db-cluster-from-snapshot \
    --db-cluster-identifier sample-cluster-restore \
    --snapshot-identifier sample-cluster-snapshot \
```

```
    --engine docdb \
    --kms-key-id arn:aws:kms:us-east-1:123456789012:key/SAMPLE-KMS-KEY-ID
```

Pour Windows :

```
aws docdb restore-db-cluster-from-snapshot ^
    --db-cluster-identifier sample-cluster-restore ^
    --snapshot-identifier sample-cluster-snapshot ^
    --engine docdb ^
    --kms-key-id arn:aws:kms:us-east-1:123456789012:key/SAMPLE-KMS-KEY-ID
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBCluster": {
        "AvailabilityZones": [
            "us-east-1c",
            "us-east-1b",
            "us-east-1a"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "sample-cluster-restore",
        "DBClusterParameterGroup": "default.docdb4.0",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "sample-cluster-restore.cluster-node.us-
east-1.docdb.amazonaws.com",
        "ReaderEndpoint": "sample-cluster-restore.cluster-node.us-
east-1.docdb.amazonaws.com",
        "MultiAZ": false,
        "Engine": "docdb",
        "EngineVersion": "4.0.0",
        "Port": 27017,
        "MasterUsername": "<master-user>",
        "PreferredBackupWindow": "02:00-02:30",
        "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcdefgh",
                "Status": "active"
            }
        ],
```

```
        "HostedZoneId": "ABCDEFGHIJKLM",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/<sample-key-id>",
        "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-cluster-
restore",
        "AssociatedRoles": [],
        "ClusterCreateTime": "2020-04-01T01:43:40.871Z",
        "DeletionProtection": true
    }
}
```

Lorsque le statut du cluster est disponible, créez au moins une instance pour le cluster.

Pour Linux, macOS ou Unix :

```
aws docdb create-db-instance \
    --db-cluster-identifier sample-cluster-restore  \
    --db-instance-identifier sample-cluster-restore-instance \
    --availability-zone us-east-1b \
    --promotion-tier 2 \
    --db-instance-class db.r5.large \
    --engine docdb
```

Pour Windows :

```
aws docdb create-db-instance ^
    --db-cluster-identifier sample-cluster-restore  ^
    --db-instance-identifier sample-cluster-restore-instance ^
    --availability-zone us-east-1b ^
    --promotion-tier 2 ^
    --db-instance-class db.r5.large ^
    --engine docdb
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "sample-cluster-restore-instance",
        "DBInstanceClass": "db.r5.large",
        "Engine": "docdb",
        "DBInstanceStatus": "creating",
        "PreferredBackupWindow": "02:00-02:30",
```

```
        "BackupRetentionPeriod": 1,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcdefgh",
                "Status": "active"
            }
        ],
        "AvailabilityZone": "us-west-2b",
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-6242c31a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-abcdefgh",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    ...
                }
            ]
        },
        "PreferredMaintenanceWindow": "fri:09:43-fri:10:13",
        "PendingModifiedValues": {},
        "EngineVersion": "4.0.0",
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "DBClusterIdentifier": "sample-cluster-restore",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/<sample-key-id>",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "CACertificateIdentifier": "rds-ca-2019",
        "PromotionTier": 2,
        "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-cluster-
restore-instance"
    }
}
```

Restaurez à un instant dans le passé

Vous pouvez restaurer un cluster à tout moment compris dans la période de conservation des sauvegardes du cluster à l'aide de la touche AWS Management Console ou AWS Command Line Interface (AWS CLI).

> **Note**
>
> Vous ne pouvez pas point-in-time restaurer un cluster 3.6 vers un cluster 4.0, mais vous pouvez migrer d'une version de cluster à une autre. Pour plus d'informations, consultez la section concernant Migration vers Amazon DocumentDB.

Gardez ce qui suit à l'esprit lorsque vous restaurez un cluster à un instant dans le passé.

- Le nouveau cluster est créé avec la même configuration que le cluster source, mais il est créé avec le groupe de paramètres par défaut. Pour définir le groupe de paramètres du nouveau cluster selon le groupe de paramètres du cluster source, modifiez le cluster une fois qu'il est disponible. Pour plus d'informations sur la modification d'un cluster, consultez Modification d'un cluster Amazon DocumentDB.

Using the AWS Management Console

Vous pouvez restaurer un cluster dans les point-in-time limites de sa période de conservation des sauvegardes en effectuant les opérations suivantes à l'aide du AWS Management Console.

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters. Dans la liste des clusters, choisissez le bouton sur la gauche du cluster que vous voulez restaurer.

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans le menu Actions, sélectionnez Restore to point in time (Restaurer à un instant dans le passé).
4. Renseignez la section Restore time (Heure de la restauration), qui spécifie la date et l'heure de la restauration.

   a. Date de restauration : choisissez ou entrez une date comprise entre l'heure de restauration la plus ancienne et l'heure de restauration la plus récente.

   b. Heure de restauration : choisissez ou entrez les heures, minutes et secondes comprises entre l'heure de restauration la plus ancienne et l'heure de restauration la plus récente.
5. Renseignez la section Configuration.

   a. Identifiant du cluster : acceptez l'identifiant par défaut ou entrez l'identifiant de votre choix.

      Contraintes d'attribution de nom relatives à un cluster :

      - La longueur est de [1 à 63] lettres, chiffres ou traits d'union.
      - Le premier caractère doit être une lettre.
      - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
      - Doit être unique pour tous les clusters d'Amazon RDS, Neptune et Amazon Compte AWS DocumentDB par région.

   b. Classe d'instance : dans la liste déroulante, choisissez la classe d'instance que vous souhaitez pour les instances du cluster.

   c. Nombre d'instances : dans la liste déroulante, choisissez le nombre d'instances que vous souhaitez créer lors de la restauration du cluster.
6. Pour la configuration du stockage en cluster, choisissez une option de stockage.

   > **Note**
   >
   > La configuration de stockage optimisée pour les E/S d'Amazon DocumentDB n'est disponible que sur la version 5.0 du moteur Amazon DocumentDB.
7. Facultatif. Pour configurer les paramètres réseau, les options de cluster, et activer les exportations de journaux, choisissez Show advanced settings (Afficher les paramètres avancés), puis renseignez les sections suivantes. Dans le cas contraire, passez à l'étape suivante.

- Network settings (Paramètres réseau)
  1. Virtual Private Cloud (VPC) : dans la liste déroulante, choisissez le VPC que vous souhaitez utiliser pour ce cluster.
  2. Groupe de sous-réseaux : dans la liste déroulante, choisissez le groupe de sous-réseaux pour ce cluster.
  3. Groupes de sécurité VPC : dans la liste déroulante, choisissez les groupes de sécurité VPC pour ce cluster.

- Options de cluster
  1. Port : acceptez le port par défaut (27017) ou utilisez les flèches haut et bas pour définir le port de communication avec ce cluster.

- Exportations des journaux
  1. Journaux d'audit : sélectionnez cette option pour activer l'exportation des journaux d'audit vers Amazon CloudWatch Logs. Si vous sélectionnez cette option, vous devez activer `audit_logs` dans le groupe de paramètres personnalisés du cluster. Pour plus d'informations, consultez Audit des événements Amazon DocumentDB.
  2. Journaux du profileur : sélectionnez cette option pour activer l'exportation des journaux du profileur d'opérations vers Amazon CloudWatch Logs. Si vous sélectionnez cette option, vous devez également modifier les paramètres suivants dans le groupe de paramètres personnalisés du cluster :
     - `profiler`— Réglé sur`enabled`.
     - `profiler_threshold_ms`— Définissez une valeur pour `[0-INT_MAX]` définir le seuil pour les opérations de profilage.
     - `profiler_sampling_rate`— Définissez une valeur pour `[0.0-1.0]` définir le pourcentage d'opérations lentes par rapport au profil.

     Pour plus d'informations, consultez Profilage des opérations Amazon DocumentDB.
  3. Journaux du profileur — Exportez les journaux du profileur vers Amazon CloudWatch
  4. Rôle IAM : dans la liste déroulante, sélectionnez RDS Service Linked Role.
- Balises

1. Ajouter une étiquette — Dans le champ Clé, entrez le nom de la balise pour votre cluster. Dans la zone Value (Valeur), entrez la valeur de la balise, si vous le souhaitez. Les balises sont utilisées avec les politiques AWS Identity and Access Management (IAM) pour gérer l'accès aux ressources Amazon DocumentDB et pour contrôler les actions qui peuvent être appliquées aux ressources.

- Deletion protection (Protection contre la suppression)

1. Activer la protection contre la suppression : protège le cluster contre toute suppression accidentelle. Lorsque cette option est activée, vous ne pouvez pas supprimer le cluster.

8. Pour restaurer le cluster, choisissez Create cluster (Créer un cluster). Vous pouvez également choisir Cancel (Annuler) pour annuler l'opération.

Using the AWS CLI

Pour restaurer un cluster à un instant dans le passé avec la période de rétention des sauvegardes de l'instantané, utilisez l'opération `restore-db-cluster-to-point-in-time` avec les paramètres suivants.

- **`--db-cluster-identifier`**— Obligatoire. Il faut créer le nom du nouveau cluster. Ce cluster ne peut pas exister avant l'opération. La valeur du paramètre doit respecter les contraintes suivantes.

  Contraintes d'attribution de nom relatives à un cluster :

  - La longueur est de [1 à 63] lettres, chiffres ou traits d'union.
  - Le premier caractère doit être une lettre.
  - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
  - Doit être unique pour tous les clusters d'Amazon RDS, Neptune et Amazon Compte AWS DocumentDB par région.

- **`--restore-to-time`**— Date et heure UTC auxquelles le cluster doit être restauré. Par exemple, `2018-06-07T23:45:00Z`.

  Contraintes de temps :

  - Elles doivent se situer avant l'heure de restauration la plus récente pour le cluster.
  - Cela doit être indiqué si le paramètre `--use-latest-restorable-time` n'est pas fourni.
  - Cela ne peut pas être spécifié lorsque le paramètre `--use-latest-restorable-time` est `true`.

  - Cela ne peut pas être spécifié lorsque la valeur du paramètre `--restore-type` est `copy-on-write`.
- **`--source-db-cluster-identifier`**— Nom du cluster source à partir duquel effectuer la restauration. Ce cluster doit exister et être disponible.
- **`--use-latest-restorable-time`**ou **`--no-use-latest-restorable-time`** — S'il faut restaurer à l'heure de sauvegarde restaurable la plus récente. Cela ne doit pas être indiqué si le paramètre `--restore-to-time` est fourni.
- **`--storage-type standard | iopt1`**— Facultatif. Par défaut: `standard`.

L' AWS CLI opération restaure `restore-db-cluster-to-point-in-time` uniquement le cluster, pas les instances de ce cluster. Vous devez invoquer l'opération `create-db-instance` pour créer des instances pour le cluster restauré, en spécifiant l'identifiant du cluster restauré dans `--db-cluster-identifier`. Vous pouvez créer des instances de bases de données uniquement après la fin de l'opération `restore-db-cluster-to-point-in-time` et lorsque le cluster restauré est disponible.

Example

L'exemple suivant crée `sample-cluster-restored` à partir de l'instantané `sample-cluster-snapshot` vers la dernière heure de restauration la plus récente.

Pour Linux, macOS ou Unix :

```
aws docdb restore-db-cluster-to-point-in-time \
    --db-cluster-identifier sample-cluster-restored \
    --source-db-cluster-identifier sample-cluster-snapshot \
    --use-latest-restorable-time
```

Pour Windows :

```
aws docdb restore-db-cluster-to-point-in-time ^
    --db-cluster-identifier sample-cluster-restored ^
    --source-db-cluster-identifier sample-cluster-snapshot ^
    --use-latest-restorable-time
```

Example

L'exemple suivant crée `sample-cluster-restored` à partir de l'instantané `sample-cluster-snapshot` du 11 décembre 2018 à 03 h 15 (UTC), inclus dans la période de rétention des sauvegardes du `sample-cluster`.

Pour Linux, macOS ou Unix :

```
aws docdb restore-db-cluster-to-point-in-time \
    --db-cluster-identifier sample-cluster-restore \
    --source-db-cluster-identifier sample-cluster \
    --restore-to-time 2020-05-12T03:15:00Z
```

Pour Windows :

```
aws docdb restore-db-cluster-to-point-in-time ^
    --db-cluster-identifier sample-cluster-restore ^
    --source-db-cluster-identifier sample-cluster ^
    --restore-to-time 2020-05-12T03:15:00Z
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBCluster": {
        "AvailabilityZones": [
            "us-east-1c",
            "us-west-2b",
            "us-west-2a"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "sample-cluster-restored",
        "DBClusterParameterGroup": "sample-parameter-group",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "sample-cluster-restored.node.us-east-1.docdb.amazonaws.com",
        "ReaderEndpoint": "sample-cluster-restored.node.us-
east-1.docdb.amazonaws.com",
        "MultiAZ": false,
        "Engine": "docdb",
        "EngineVersion": "4.0.0",
        "Port": 27017,
```

```
        "MasterUsername": "master-user",
        "PreferredBackupWindow": "02:00-02:30",
        "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abc0123",
                "Status": "active"
            }
        ],
        "HostedZoneId": "ABCDEFGHIJKLM",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID^>:key/sample-key",
        "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-cluster-
restored",
        "AssociatedRoles": [],
        "ClusterCreateTime": "2020-04-24T20:14:36.713Z",
        "DeletionProtection": false
    }
}
```

Suppression d'un instantané de cluster

Un instantané manuel est une sauvegarde complète qui est supprimée uniquement lorsque vous la supprimez manuellement à l'aide du AWS Management Console ou AWS CLI. Vous ne pouvez pas supprimer manuellement des instantanés automatiques, car ils sont uniquement supprimés lorsque la période de rétention de l'instantané arrive à expiration ou que vous supprimez le cluster de l'instantané.

Using the AWS Management Console

Pour supprimer un instantané de cluster manuel à l'aide du AWS Management Console, procédez comme suit.

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Snapshots (Instantanés).

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans la liste des instantanés, choisissez le bouton sur la gauche de l'instantané que vous voulez supprimer. Le type d'instantané doit être manuel.

   1. Vous pouvez savoir si le type de l'instantané est manuel en vérifiant s'il est répertorié comme `manual` ou `automatic` dans la colonne Type .

4. Dans le menu Actions, choisissez Delete (Supprimer). Si l'option Supprimer n'est pas disponible, vous avez sans doute choisi un instantané automatique.
5. Sur l'écran de confirmation de la suppression, choisissez Supprimer pour supprimer l'instantané. Pour conserver les instantanés, choisissez Annuler.

Using the AWS CLI

Un instantané de cluster manuel Amazon DocumentDB est une sauvegarde complète que vous pouvez supprimer manuellement à l'aide du. AWS CLI Vous ne pouvez pas supprimer manuellement un instantané automatique.

Pour supprimer un instantané de cluster manuel à l'aide de AWS CLI, utilisez l'`delete-db-cluster-snapshot`opération avec les paramètres suivants.

Paramètres

- **`--db-cluster-snapshot-identifier`** — Obligatoire. Le nom du projet de l'instantané manuel à supprimer.

L'exemple suivant supprime l'instantané du cluster `sample-cluster-snapshot`.

Pour Linux, macOS ou Unix :

```
aws docdb delete-db-cluster-snapshot \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Pour Windows :

```
aws docdb delete-db-cluster-snapshot ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

La sortie de cette opération répertorie les détails de l'instantané de cluster que vous avez supprimé.

Gestion des ressources Amazon DocumentDB

Ces sections couvrent les différents composants et leurs tâches associées pour gérer votre implémentation d'Amazon DocumentDB (avec compatibilité MongoDB).

Rubriques

Présentation des tâches opérationnelles d'Amazon DocumentDB

Cette rubrique présente les tâches opérationnelles pour votre cluster Amazon DocumentDB (avec compatibilité MongoDB) et la façon d'effectuer ces tâches à l'aide de laAWS CLI.

Rubriques

- Création d'un instantané de cluster
- Restaurer à partir d'un instantané
- Suppression d'une instance dans un cluster
- Suppression d'un cluster

Ajouter un réplica à un cluster Amazon DocumentDB

Une fois que vous avez créé l'instance principale pour votre cluster Amazon DocumentDB, vous pouvez ajouter un ou plusieursrépliques. Un réplica est une instance en lecture seule qui a deux finalités :

- Evolutivité— Si un grand nombre de clients doit accéder simultanément, vous pouvez ajouter plus de réplicas pour la lecture de dimensionnement.
- Haute disponibilité— Si l'instance principale échoue, Amazon DocumentDB bascule automatiquement vers une instance de réplica et la désigne en tant que nouvelle instance principale. Si un réplica échoue, d'autres instances dans le cluster peuvent tout de même servir des demandes jusqu'à ce que le nœud en échec puisse être récupéré.

Chaque cluster Amazon DocumentDB peut prendre en charge jusqu'à 15 réplicas.

> **Note**
>
> Pour bénéficier d'une tolérance aux pannes maximale, vous devez déployer les réplicas dans des zones de disponibilité distinctes. Cette configuration est l'assurance que votre cluster Amazon DocumentDB peut continuer à fonctionner, même si une zone de disponibilité entière devient indisponible.

L'exemple suivant de la AWS CLI montre comment ajouter un nouveau réplica. Le paramètre `--availability-zone` place le réplica dans la zone de disponibilité spécifiée.

```
aws docdb create-db-instance \
    --db-instance-identifier sample-instance \
    --db-cluster-identifier sample-cluster \
    --engine docdb \
    --db-instance-class db.r5.large \
    --availability-zone us-east-1a
```

Description des clusters et des instances

Procédez comme suit :AWS CLIrépertorie tous les clusters Amazon DocumentDB dans une région. Pour certaines fonctionnalités de gestion telles que la gestion du cycle de vie d'un cluster et d'une instance, Amazon DocumentDB exploite la technologie opérationnelle partagée avec Amazon RDS. `LefilterName=engine,Values=docdb`renvoie uniquement les clusters Amazon DocumentDB.

Pour plus d'informations sur la description et la modification des clusters, veuillez consulter Cycle de vie du cluster Amazon DocumentDB.

```
aws docdb describe-db-clusters --filter Name=engine,Values=docdb
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBClusters": [
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-1",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        },
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-2",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
```

```
            "Status": "available",
            ...
        },
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-3",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        }
    ]
}
```

Procédez comme suit :AWS CLIrépertorie les instances d'un cluster Amazon DocumentDB. Pour plus d'informations sur la description et la modification des clusters, veuillez consulter Cycle de vie des instances Amazon DocumentDB.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].[DBClusterMembers]'
```

Le résultat se présente comme suit. Il existe deux sections dans ce résultat. L'instance principale est `sample-instance-1` (`"IsClusterWriter": true`). Il existe également une instance de réplica, `sample-instance2` (`"IsClusterWriter: false"`).

```
[
    [
        [
            {
                "DBInstanceIdentifier": "sample-instance-1",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            },
            {
                "DBInstanceIdentifier": "sample-cluster-2",
```

```
                "IsClusterWriter": false,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ]
    ]
]
```

Création d'un instantané de cluster

UNInstantané de clusterest une sauvegarde complète des données dans votre cluster Amazon DocumentDB. Lorsque l'instantané est en cours de création, Amazon DocumentDB lit vos données directement à partir du volume de cluster. Pour cette raison, vous pouvez créer un instantané, même aucune instance n'est exécutée à ce moment-là dans votre cluster. Le temps nécessaire à la création d'un instantané varie en fonction de la taille du volume de votre cluster.

Amazon DocumentDB prend en charge les sauvegardes automatiques, exécutées chaque jour pendant la créneau de sauvegarde préférée, soit une période de 30 minutes au cours de la journée. L'exemple de la AWS CLI suivant montre comment afficher le créneau de sauvegarde pour votre cluster :

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].PreferredBackupWindow'
```

Le résultat montre le créneau de sauvegarde (au format UTC) :

```
[
    "00:18-00:48"
]
```

Vous pouvez définir la créneau de sauvegarde lors de la création de votre cluster Amazon DocumentDB. Vous pouvez également modifier le créneau de sauvegarde, comme l'illustre l'exemple suivant. Si vous ne définissez pas de créneau de sauvegarde, Amazon DocumentDB en attribue automatiquement un à votre cluster.

```
aws docdb modify-db-cluster \
    --db-cluster-identifier sample-cluster \
```

```
    --preferred-backup-window "02:00-02:30"
```

Outre les sauvegardes automatiques, vous pouvez créer manuellement un instantané de cluster à tout moment. En faisant cela, vous spécifiez le cluster que vous souhaitez sauvegarder, et un nom unique pour votre instantané, afin de pouvoir restaurer à partir de ce dernier ultérieurement.

L'exemple AWS CLI suivant montre comment créer un instantané de vos données.

```
aws docdb create-db-cluster-snapshot \
    --db-cluster-identifier sample-cluster \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Restaurer à partir d'un instantané

Vous pouvez restaurer un instantané de cluster vers un nouveau cluster Amazon DocumentDB. Pour ce faire, vous fournissez le nom de l'instantané et le nom d'un nouveau cluster. Vous ne pouvez pas effectuer une restauration à partir d'un instantané vers un cluster existant ; au lieu de cela, Amazon DocumentDB crée un cluster lorsque vous effectuez la restauration, puis le remplit avec vos données d'instantané.

L'exemple suivant montre tous les instantanés pour le cluster `sample-cluster`.

```
aws docdb describe-db-cluster-snapshots \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusterSnapshots[*].[DBClusterSnapshotIdentifier,SnapshotType,Status]'
```

Le résultat se présente comme suit. Un instantané manuel est créé manuellement, tandis qu'un instantané automatisé est créé par Amazon DocumentDB dans le créneau de sauvegarde du cluster.

```
[
        "sample-cluster-snapshot",
        "manual",
        "available"
    ],
    [
        "rds:sample-cluster",
        "automated",
        "available"
    ]
]
```

L'exemple suivant montre comment restaurer un cluster Amazon DocumentDB à partir d'un instantané.

```
aws docdb restore-db-cluster-from-snapshot \
    --engine docdb \
    --db-cluster-identifier new-sample-cluster \
    --snapshot-identifier sample-cluster-snapshot
```

Le nouveau cluster n'a aucune instance associée. Par conséquent, si vous souhaitez interagir avec le cluster, vous devez ajouter une instance à ce dernier.

```
aws docdb create-db-instance \
    --db-instance-identifier new-sample-instance \
    --db-instance-class db.r5.large \
    --engine docdb \
    --db-cluster-identifier new-sample-cluster
```

Vous pouvez utiliser les opérations AWS CLI suivantes pour surveiller l'avancement de la création de l'instance et du cluster. Lorsque les statuts du cluster et de l'instance sont disponibles, vous pouvez vous connecter au point de terminaison du nouveau cluster et accéder à vos données.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier new-sample-cluster  \
    --query 'DBClusters[*].[Status,Endpoint]'
```

```
aws docdb describe-db-instances \
    --db-instance-identifier new-sample-instance \
    --query 'DBInstances[*].[DBInstanceStatus]'
```

Suppression d'une instance dans un cluster

Amazon DocumentDB stocke toutes vos données dans le volume de cluster. Les données restent dans ce volume de cluster, même si vous supprimez toutes les instances de votre cluster. Si vous avez besoin d'accéder aux données à nouveau, vous pouvez ajouter une instance au cluster à tout moment, et reprendre votre activité là où vous l'aviez laissée.

L'exemple suivant montre comment supprimer une instance à partir de votre cluster Amazon DocumentDB.

```
aws docdb delete-db-instance \
```

```
    --db-instance-identifier sample-instance
```

Suppression d'un cluster

Avant de pouvoir supprimer un cluster Amazon DocumentDB, vous devez tout d'abord supprimer toutes ses instances. L'exemple AWS CLI suivant renvoie les informations suivantes sur les instances d'un cluster. Si cette opération renvoie des identifiants d'instance, vous devez supprimer chacune des instances. Pour plus d'informations, consultez Suppression d'une instance dans un cluster.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].DBClusterMembers[*].DBInstanceIdentifier'
```

Quand il ne reste plus d'instances, vous pouvez supprimer le cluster. A ce moment-là, vous devez choisir l'une des options suivantes :

- Créer un instantané final— Capturez toutes les données du cluster dans un instantané, ce qui vous permet de recréer ultérieurement une nouvelle instance avec ces données. L'exemple suivant illustre la marche à suivre :

  ```
  aws docdb delete-db-cluster \
      --db-cluster-identifier sample-cluster \
      --final-db-snapshot-identifier sample-cluster-snapshot
  ```

- Ignorer l'instantané final— Supprimer définitivement toutes les données du cluster. Cette opération ne peut pas être annulée. L'exemple suivant illustre la marche à suivre :

  ```
  aws docdb delete-db-cluster \
      --db-cluster-identifier sample-cluster \
      --skip-final-snapshot
  ```

Présentation des clusters globaux Amazon DocumentDB

Qu'est-ce qu'un cluster mondial ?

Un cluster mondial se compose d'une région principale et d'un maximum de cinq régions secondaires en lecture seule. Vous effectuez des opérations d'écriture directement sur le cluster principal de la

région principale et Amazon DocumentDB réplique automatiquement les données vers les régions secondaires à l'aide d'une infrastructure dédiée. La latence est généralement inférieure à une seconde.

En quoi les clusters mondiaux sont-ils utiles ?

- Restauration après des pannes à l'échelle de la région : en cas de panne régionale, vous pouvez transformer l'un des clusters secondaires en cluster principal en quelques minutes, avec un objectif de temps de restauration (RTO) typique inférieur à une minute. L'objectif du point de restauration (RPO) est généralement mesuré en secondes, mais cela dépend du décalage sur le réseau au moment de la panne.
- Lectures globales avec latence locale — Si vous avez des bureaux dans le monde entier, vous pouvez utiliser un cluster mondial pour tenir à jour vos principales sources d'informations dans la région principale. Les bureaux de vos autres régions peuvent accéder aux informations de leur propre région, avec une latence locale.
- Clusters secondaires évolutifs : vous pouvez redimensionner vos clusters secondaires en ajoutant davantage d'instances en lecture seule dans une région secondaire. Le cluster secondaire est en lecture seule, il peut donc prendre en charge jusqu'à 16 instances de réplication en lecture seule au lieu de la limite habituelle de 15 pour un seul cluster.
- Réplication rapide des clusters principaux vers les clusters secondaires : la réplication effectuée par un cluster global a peu d'impact sur les performances du cluster de base de données principal. Les ressources des instances de base de données sont entièrement dévouées aux charges de travail d'application en lecture et en écriture.

Quelles sont les limites actuelles des clusters mondiaux ?

- Les clusters globaux ne sont pas pris en charge sur Amazon DocumentDB v3.6.
- Les clusters globaux ne sont pas pris en charge sur les types d'instance t3, t4g et r4.
- Les clusters mondiaux ne sont pas disponibles dans les régions suivantes : Amérique du Sud (São Paulo), Europe (Milan), Chine (Pékin) et Chine (Ningxia).
- En cas de basculement régional, vous devez promouvoir manuellement un cluster secondaire pour qu'il devienne le cluster principal et modifier votre application pour qu'elle pointe vers le nouveau cluster principal.
- Seul le cluster principal exécute les opérations d'écriture. Les clients qui effectuent des opérations d'écriture se connectent au point de terminaison du cluster principal.

- Vous pouvez avoir un maximum de cinq régions secondaires et une région principale pour votre cluster.
- Un cluster secondaire ne peut pas être arrêté. Un cluster principal ne peut pas être arrêté s'il est associé à des clusters secondaires. Seul un cluster régional qui ne possède aucun cluster secondaire peut être arrêté.
- Les répliques associées au cluster secondaire peuvent redémarrer dans certaines circonstances. Si l'instance de la région principale redémarre ou bascule, les répliques de la région secondaire redémarrent également. Le cluster est alors indisponible jusqu'à ce que toutes les répliques soient à nouveau synchronisées avec l'instance d'écriture du cluster de base de données principal. Ce comportement est normal. Assurez-vous de bien comprendre l'impact sur votre cluster global avant d'apporter des modifications à votre cluster principal.
- Vous ne pouvez pas utiliser de flux de modifications sur des clusters secondaires.

Rubriques

Guide de démarrage rapide : Global Clusters

Rubriques

Configuration

Le cluster global Amazon DocumentDB s'étend sur au moins deux. Régions AWS La région principale prend en charge un cluster composé d'une instance principale (d'écriture) et d'un maximum de quinze instances de réplique, tandis que la région secondaire gère un cluster en lecture seule

composé entièrement de seize instances de réplique au maximum. Un cluster mondial peut comporter jusqu'à cinq régions secondaires. Le tableau répertorie le nombre maximal de clusters, d'instances et de répliques autorisés dans un cluster global.

Description	Région AWS principale	Région AWS secondaire
Clusters	1	5 (maximum)
Instances de scripteur	1	0
Instances en lecture seule (répliques Amazon DocumentDB), par cluster	15 (max)	16 (total)
Instances en lecture seule (maximum autorisé, compte tenu du nombre réel de régions secondaires)	15 - s	s = nombre total d'Régions AWS secondaires

Les clusters ont les exigences spécifiques suivantes :

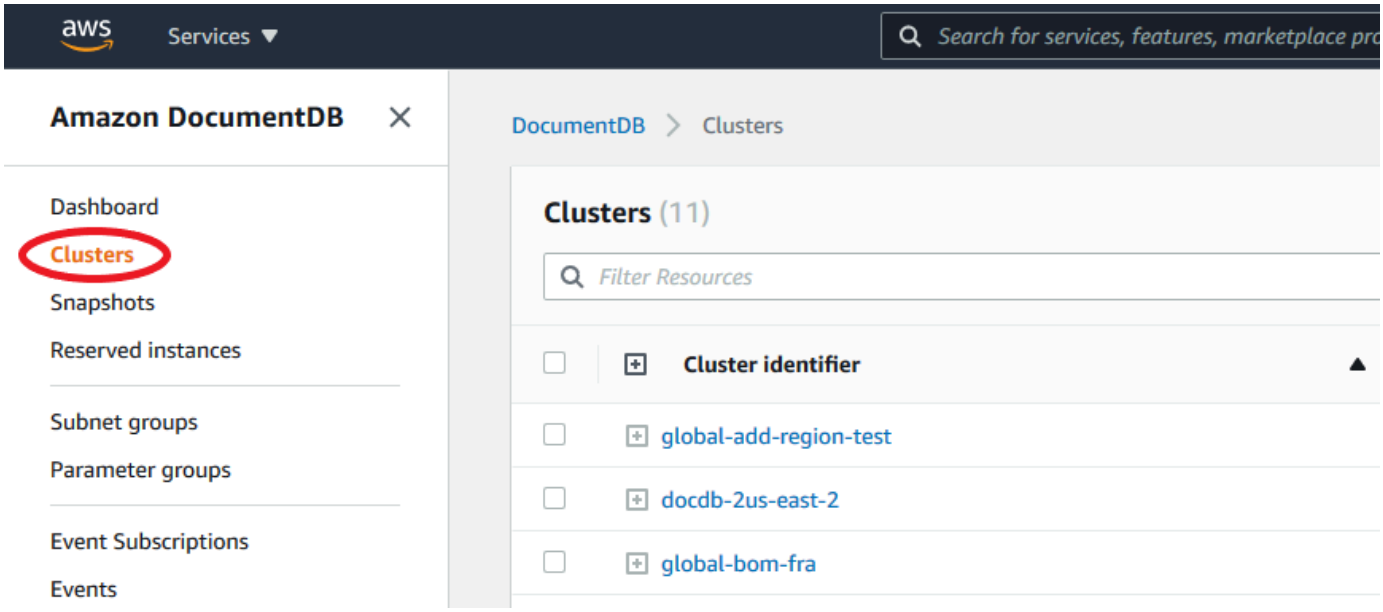
- Exigences relatives aux classes d'instance de base de données : vous ne pouvez utiliser que les classes d'`db.r6`instance `db.r5` et.
- Région AWSexigences — Le cluster principal doit se trouver dans une région, et au moins un cluster secondaire doit se trouver dans une région différente du même compte. Vous pouvez créer jusqu'à cinq clusters secondaires (en lecture seule), chacun devant se trouver dans une région différente. En d'autres termes, deux clusters ne peuvent pas se trouver dans la même région.
- Exigences en matière de dénomination — Les noms que vous choisissez pour chacun de vos clusters doivent être uniques, dans toutes les régions. Vous ne pouvez pas utiliser le même nom pour différents clusters, même s'ils se trouvent dans des régions différentes.

Création d'un cluster global Amazon DocumentDB

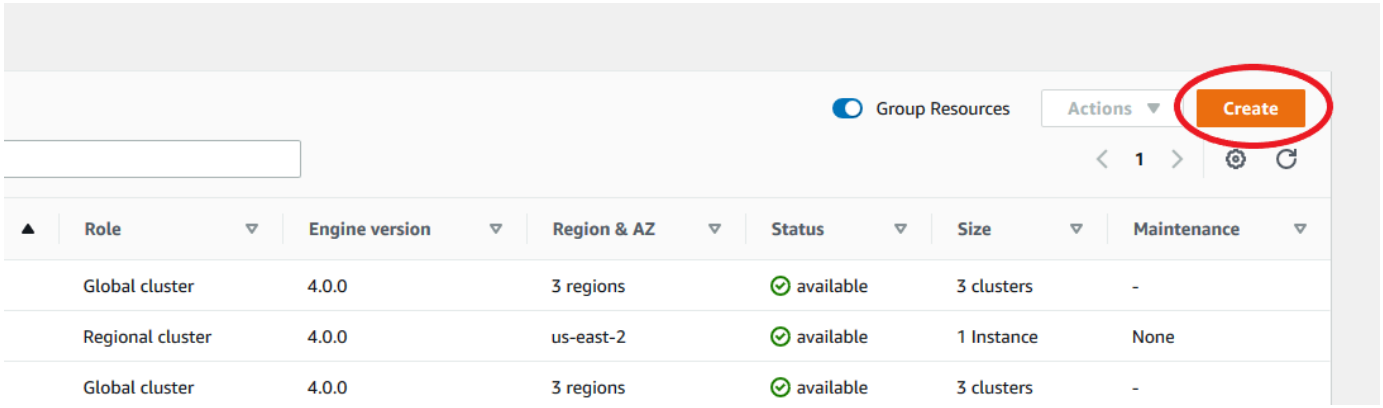
Êtes-vous prêt à créer votre premier cluster mondial ? Dans cette section, nous expliquerons comment créer un tout nouveau cluster global avec de nouveaux clusters et instances de base de données, en utilisant les instructions suivantes AWS Management Console ou AWS CLI en suivant les instructions suivantes.

Utilisation de la AWS Management Console

1. Dans leAWS Management Console, accédez à Amazon DocumentDB.
2. Lorsque vous accédez à la console Amazon DocumentDB, choisissez Clusters.

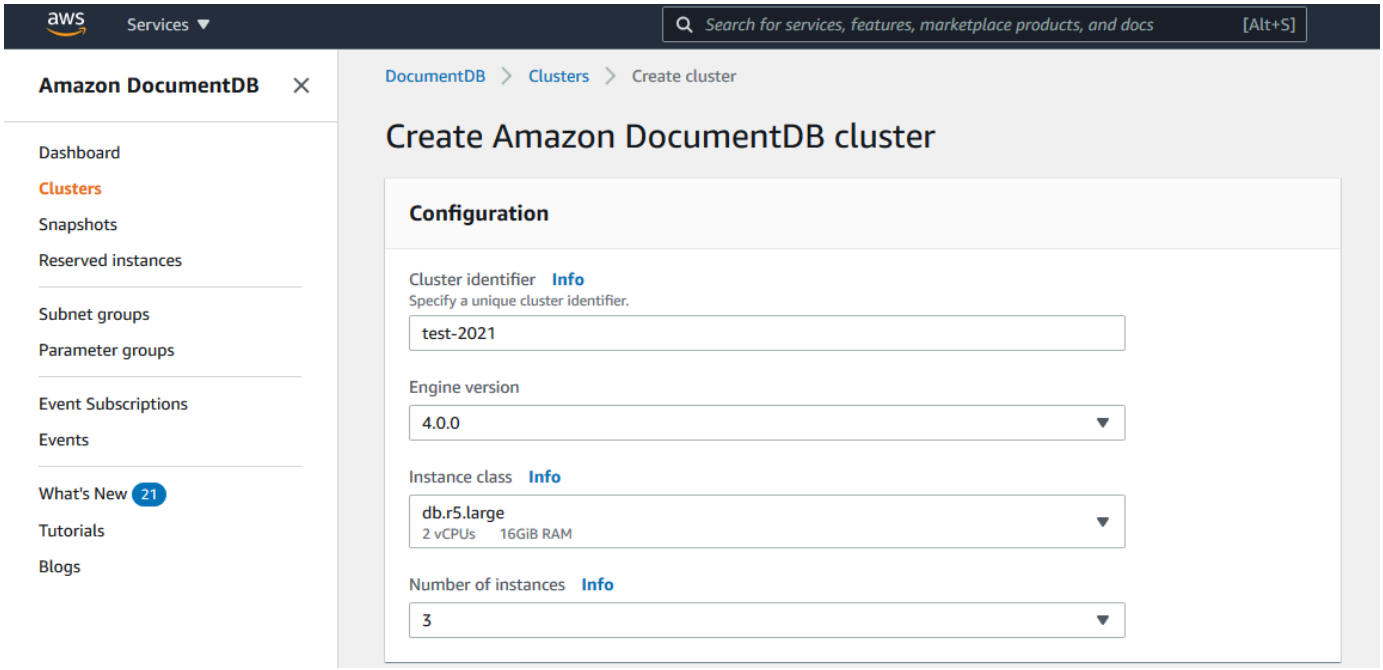


3. Choisissez Créer.

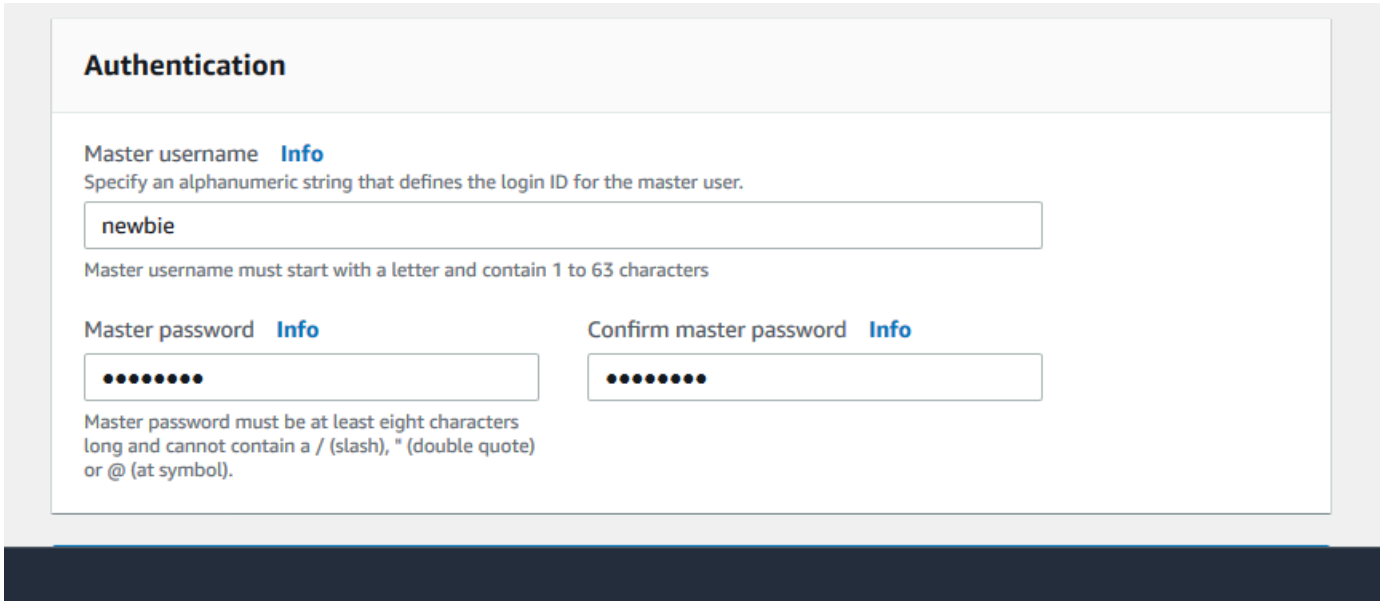


4. Remplissez la section Configuration du formulaire de création d'un cluster Amazon DocumentDB en conséquence :

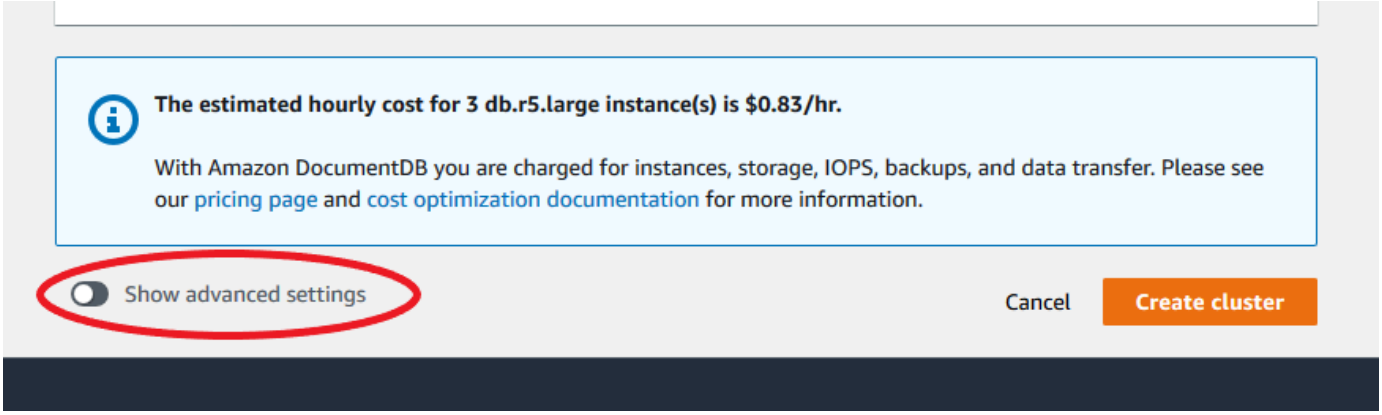
   - Identifiant de cluster : vous pouvez saisir un identifiant unique pour cette instance ou autoriser Amazon DocumentDB à fournir l'identifiant d'instance en fonction de l'identifiant de cluster.
   - Version du moteur : Choose 4.0.0
   - Classe d'instance : Choisissez db.r5.large
   - Nombre d'instances : Choisissez 3.



5. Dans la section Authentification, saisissez un nom d'utilisateur principal et un mot de passe principal.

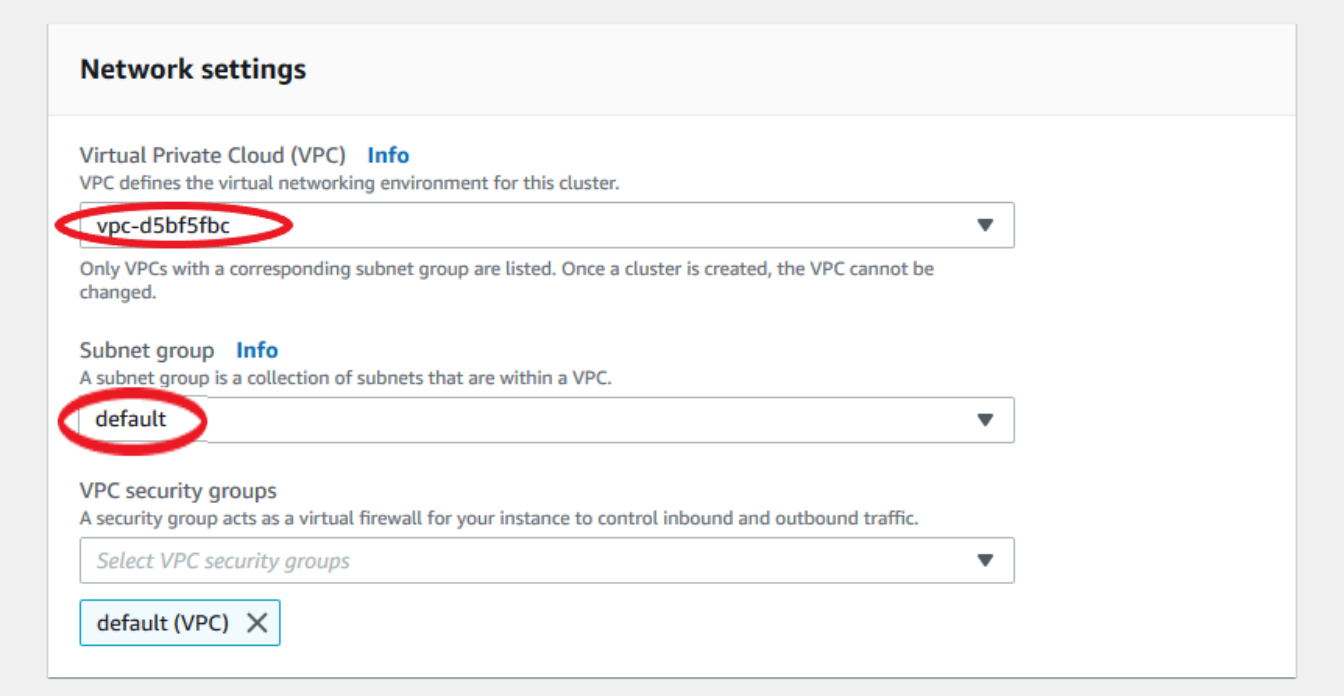


6. Choisissez Afficher les paramètres avancés.

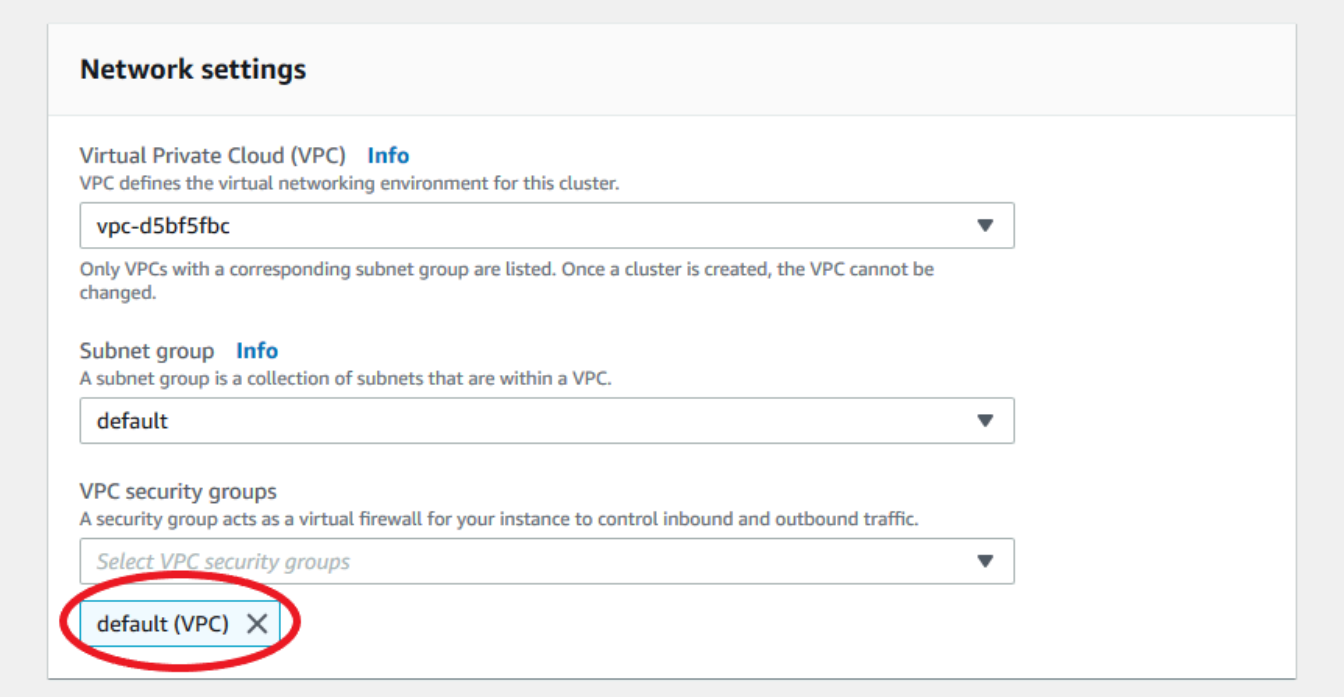


7. Dans la section Paramètres réseau :

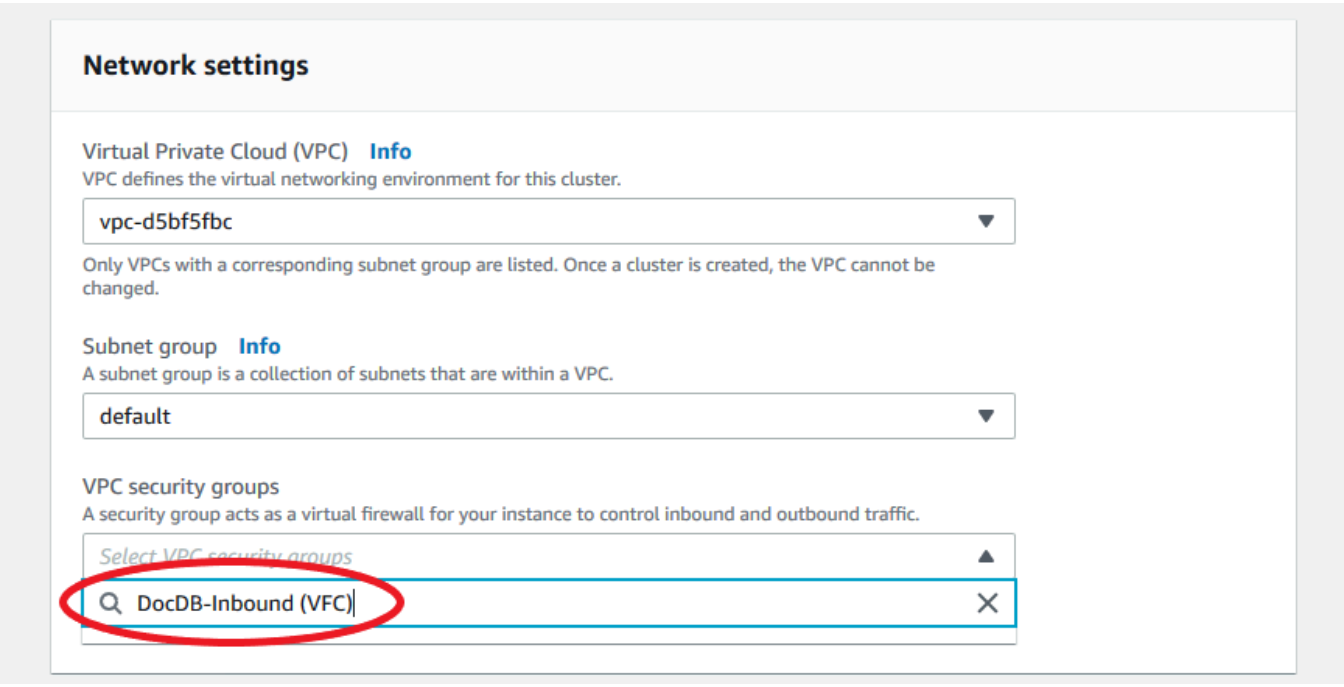
   - Conservez les options par défaut pour le cloud privé virtuel et le groupe de sous-réseaux.



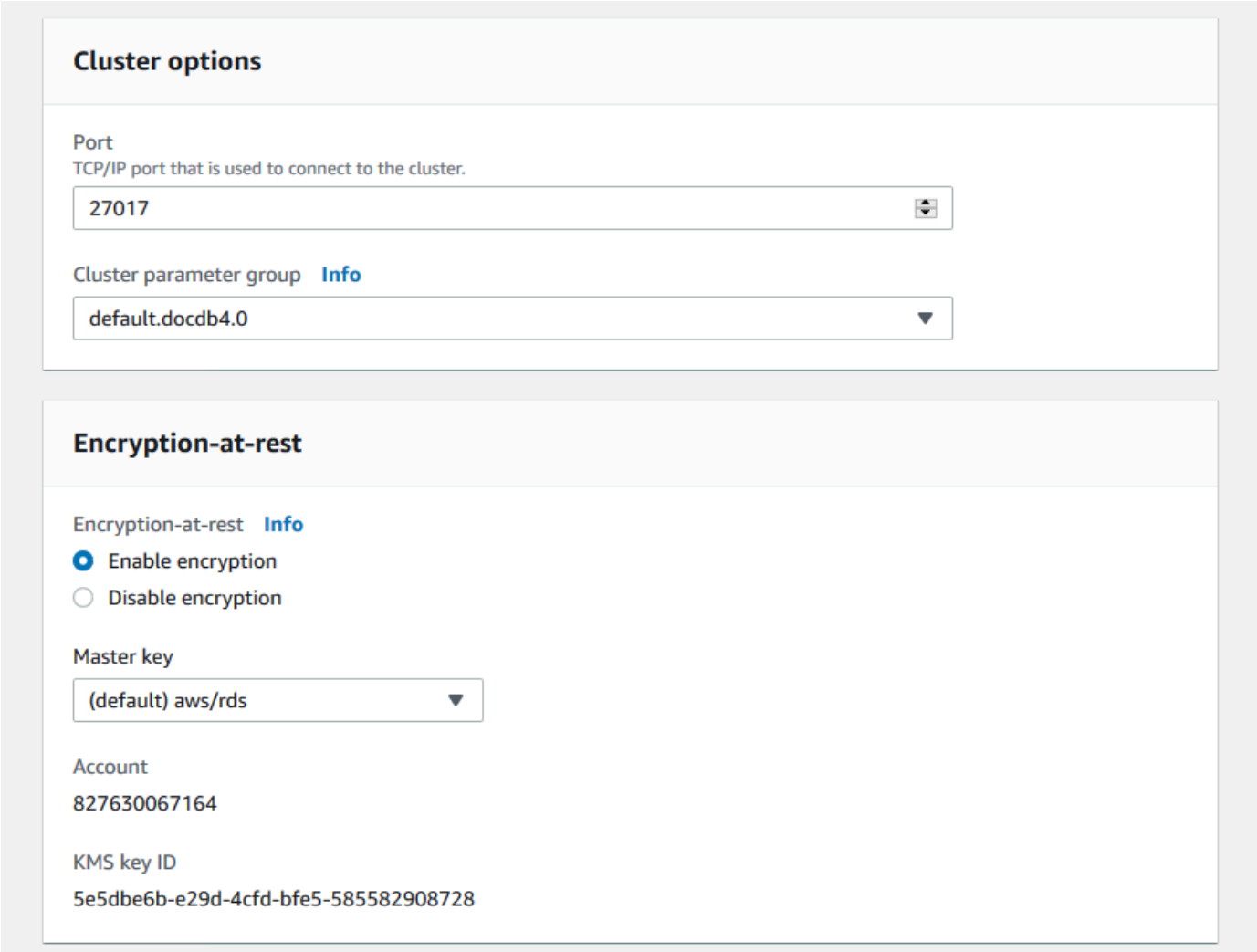
   - Pour les groupes de sécurité VPC, le VPC par défaut doit déjà être ajouté.



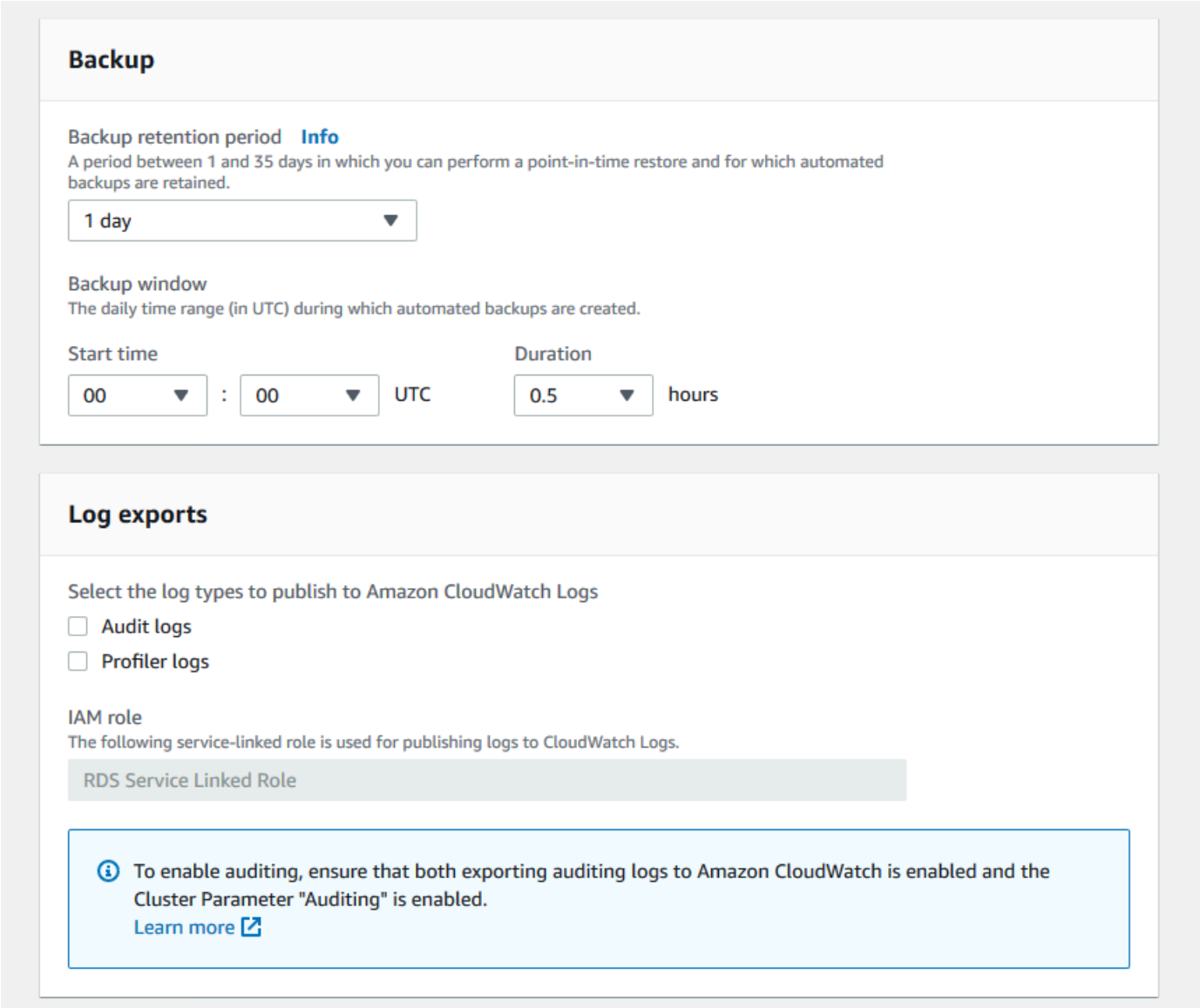
- Tapez `DocDB` dans le champ Groupes de sécurité VPC et sélectionnez DocDB-Inbound (VPC).



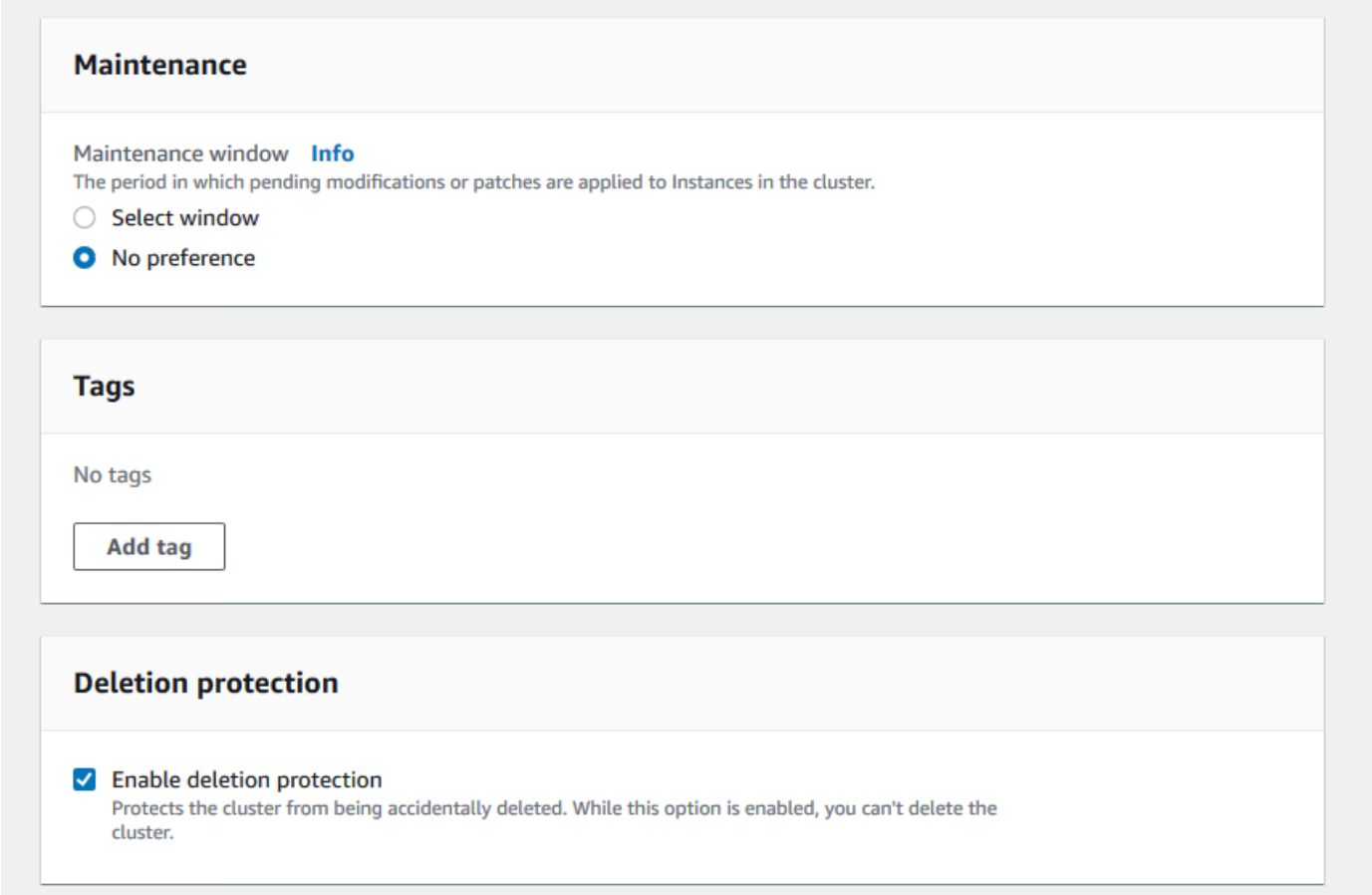
8. Pour les options de cluster et E ncryption-at-rest, conservez les sélections par défaut.



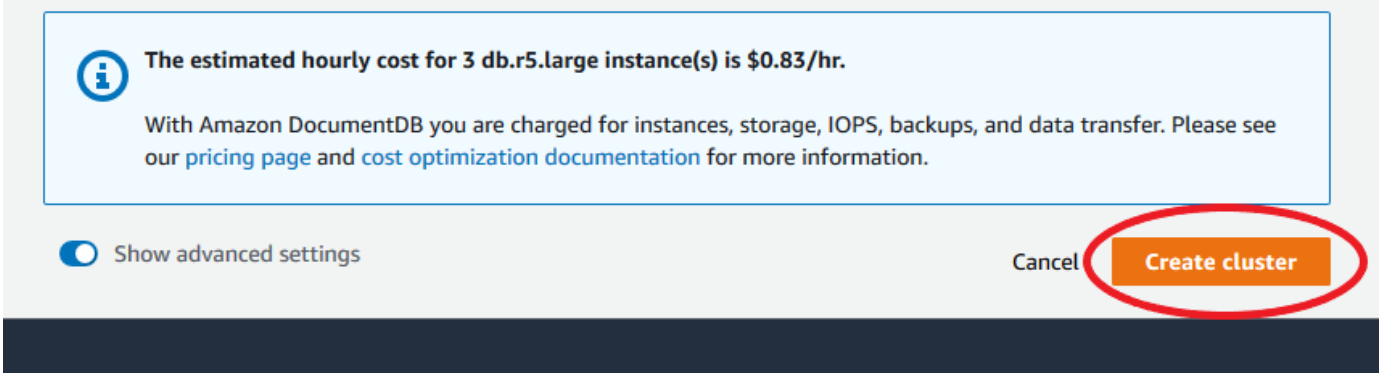
9. Pour Backup et Log Exports, conservez les sélections par défaut.



10. Pour la maintenance, les balises et la protection contre la suppression, conservez les sélections par défaut.



11. Cliquez maintenant sur le bouton Créer.



Utilisation de la AWS CLI

Pour créer un cluster régional Amazon DocumentDB, appelez le. `create-db-cluster` AWS CLI La AWS CLI commande suivante crée un cluster Amazon DocumentDB nommé. `global-cluster-id` Pour plus d'informations sur la protection contre les suppressions, consultezSuppression d'un cluster Amazon DocumentDB.

Il s'`--engine-version`agit également d'un paramètre facultatif qui utilise par défaut la dernière version majeure du moteur. La version principale actuelle du moteur est`4.0.0`. Lorsque de nouvelles versions majeures du moteur sont publiées, la version par défaut du moteur est mise à jour pour `--engine-version` refléter la dernière version du moteur principal. Par conséquent, pour les charges de travail de production, et en particulier celles qui dépendent de scripts, d'automatisation ou de AWS CloudFormation modèles, nous vous recommandons de spécifier explicitement la `--engine-version` version majeure prévue.

Si un `db-subnet-group-name` ou n'`vpc-security-group-id`est pas spécifié, Amazon DocumentDB utilisera le groupe de sous-réseaux et le groupe de sécurité Amazon VPC par défaut pour la région donnée.

Dans les exemples suivants, remplacez chaque *`espace réservé pour l'entrée utilisateur`* par vos propres informations.

Pour Linux, macOS ou Unix :

```
aws docdb create-db-cluster \
      --global-cluster-identifier global-cluster-id \
      --source-db-cluster-identifier arn:aws:rds:us-east-1:111122223333:cluster-id
```

Pour Windows :

```
aws docdb create-db-cluster ^
      --global-cluster-identifier global-cluster-id ^
      --source-db-cluster-identifier arn:aws:rds:us-east-1:111122223333:cluster-id
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "DBCluster": {
        "StorageEncrypted": false,
        "DBClusterMembers": [],
        "Engine": "docdb",
        "DeletionProtection" : "enabled",
        "ClusterCreateTime": "2018-11-26T17:15:19.885Z",
        "DBSubnetGroup": "default",
        "EngineVersion": "4.0.0",
```

```
        "MasterUsername": "masteruser",
        "BackupRetentionPeriod": 1,
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:cluster-id",
        "DBClusterIdentifier": "cluster-id",
        "MultiAZ": false,
        "DBClusterParameterGroup": "default.docdb4.0",
        "PreferredBackupWindow": "09:12-09:42",
        "DbClusterResourceId": "cluster-KQSGI4MHU4NTDDRVNLNTU7XVAY",
        "PreferredMaintenanceWindow": "tue:04:17-tue:04:47",
        "Port": 27017,
        "Status": "creating",
        "ReaderEndpoint": "cluster-id.cluster-ro-sfcrlcjcoroz.us-
east-1.docdb.amazonaws.com",
        "AssociatedRoles": [],
        "HostedZoneId": "ZNKXTT8WH85VW",
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1e"
        ],
        "Endpoint": "cluster-id.cluster-sfcrlcjcoroz.us-east-1.docdb.amazonaws.com"
    }
}
```

La création du cluster prend quelques minutes. Vous pouvez utiliser AWS Management Console ou l'AWS CLI pour surveiller l'état de votre cluster. Pour plus d’informations, consultez Surveillance de l'état d'un cluster Amazon DocumentDB.

> **Important**
>
> Lorsque vous utilisez le AWS CLI pour créer un cluster régional Amazon DocumentDB, aucune instance n'est créée. Par conséquent, vous devez créer explicitement une instance principale et tous les réplicas des instances dont vous avez besoin. Vous pouvez utiliser la console ou l'AWS CLI pour créer les instances. Pour plus d'informations, consultez Ajouter

une instance Amazon DocumentDB à un cluster et consultez le CreateDBCluster manuel Amazon DocumentDB API Reference.

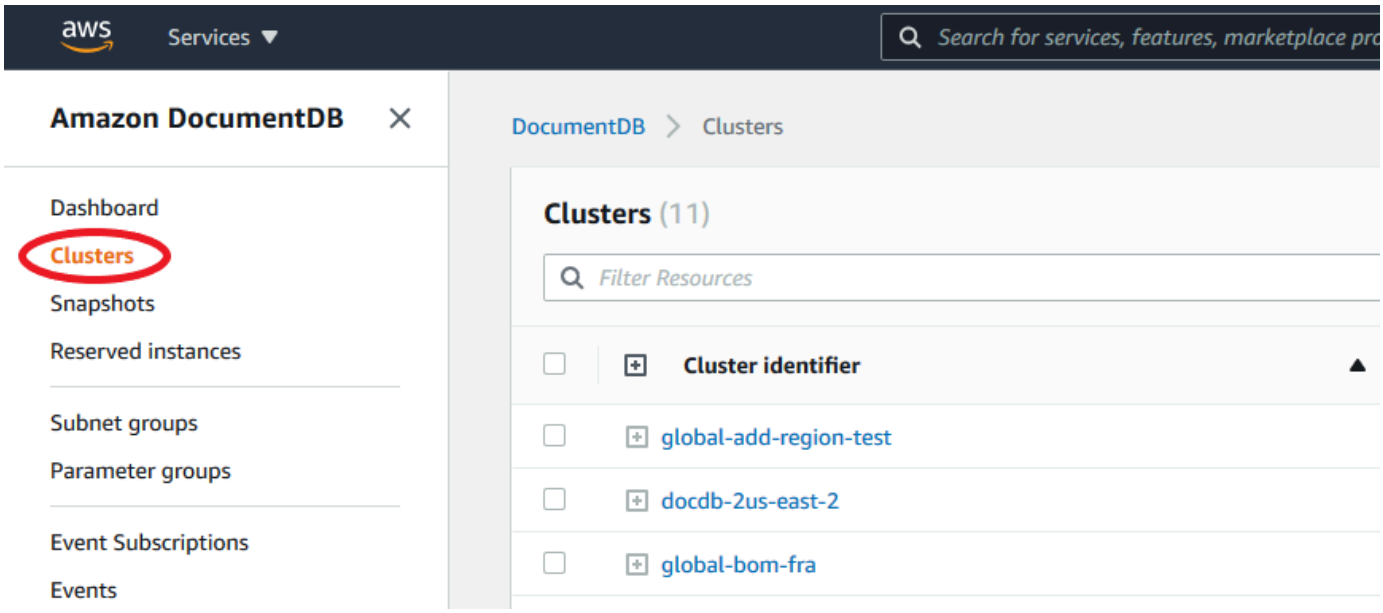
Une fois que votre cluster régional est disponible, vous pouvez ajouter un cluster secondaire dans une autre région en suivant les instructions suivantes :Ajouter un Région AWS à un cluster global Amazon DocumentDB. Lorsque vous ajoutez une région, votre cluster régional devient votre cluster principal, et vous avez un nouveau cluster secondaire dans la région que vous avez choisie.

Ajouter un Région AWS à un cluster global Amazon DocumentDB

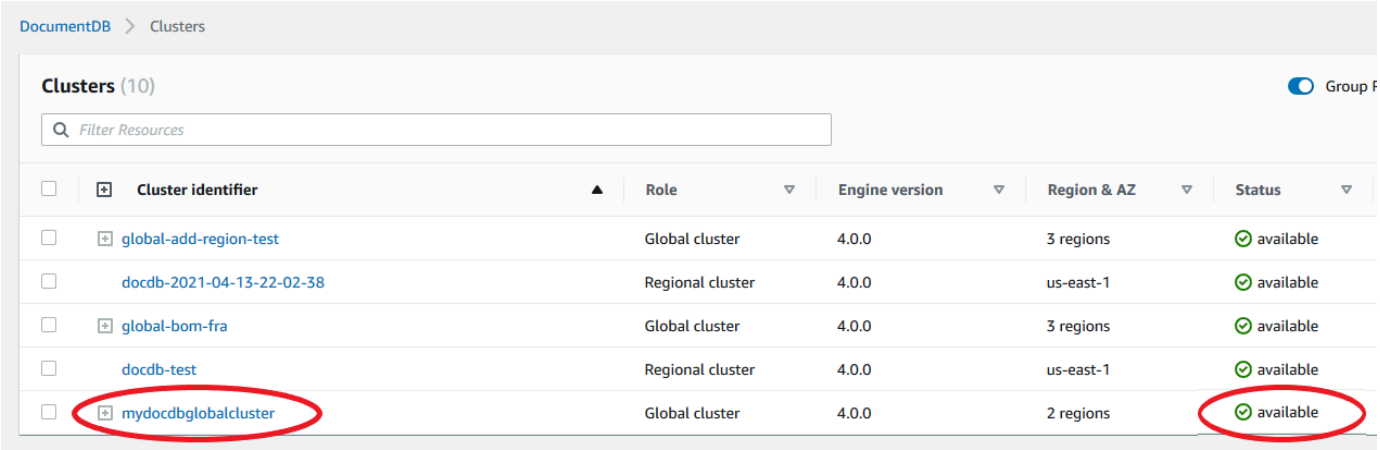
Un cluster global a besoin d'au moins un cluster secondaire dans une région différente de celle du cluster principal, et vous pouvez ajouter jusqu'à cinq clusters secondaires. Notez que pour chaque cluster secondaire que vous ajoutez, vous devez réduire d'un le nombre de répliques autorisées dans le cluster principal. Par exemple, si votre cluster global comporte cinq régions secondaires, votre cluster principal ne peut avoir que dix (au lieu de quinze) répliques. Pour plus d'informations, consultez la section Exigences de configuration d'un cluster global Amazon DocumentDB.

Utilisation de la AWS Management Console

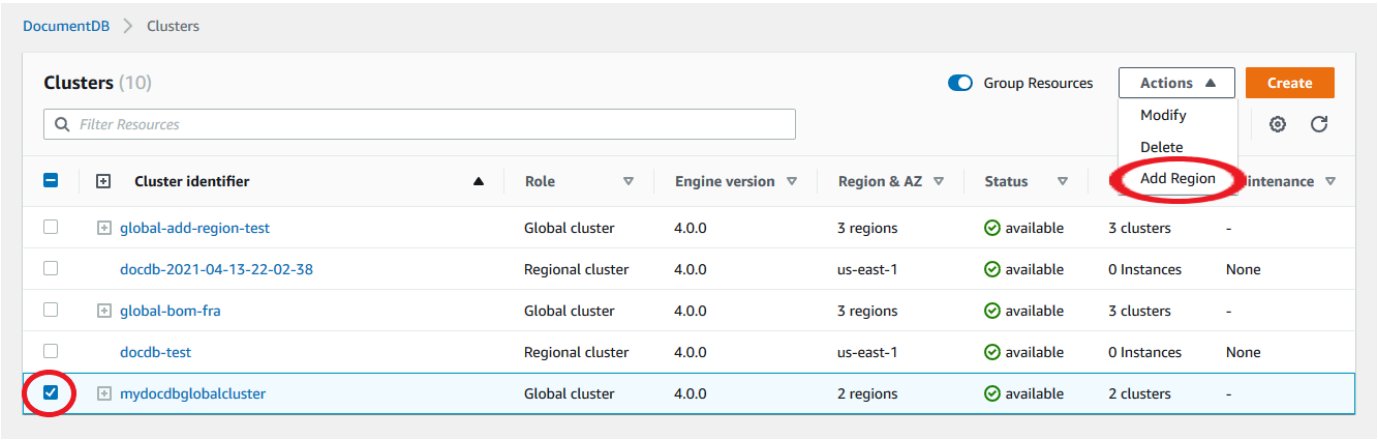
1. Connectez-vous à la console Amazon DocumentDB AWS Management Console et ouvrez-la.
2. Dans le panneau de navigation, choisissez Clusters.



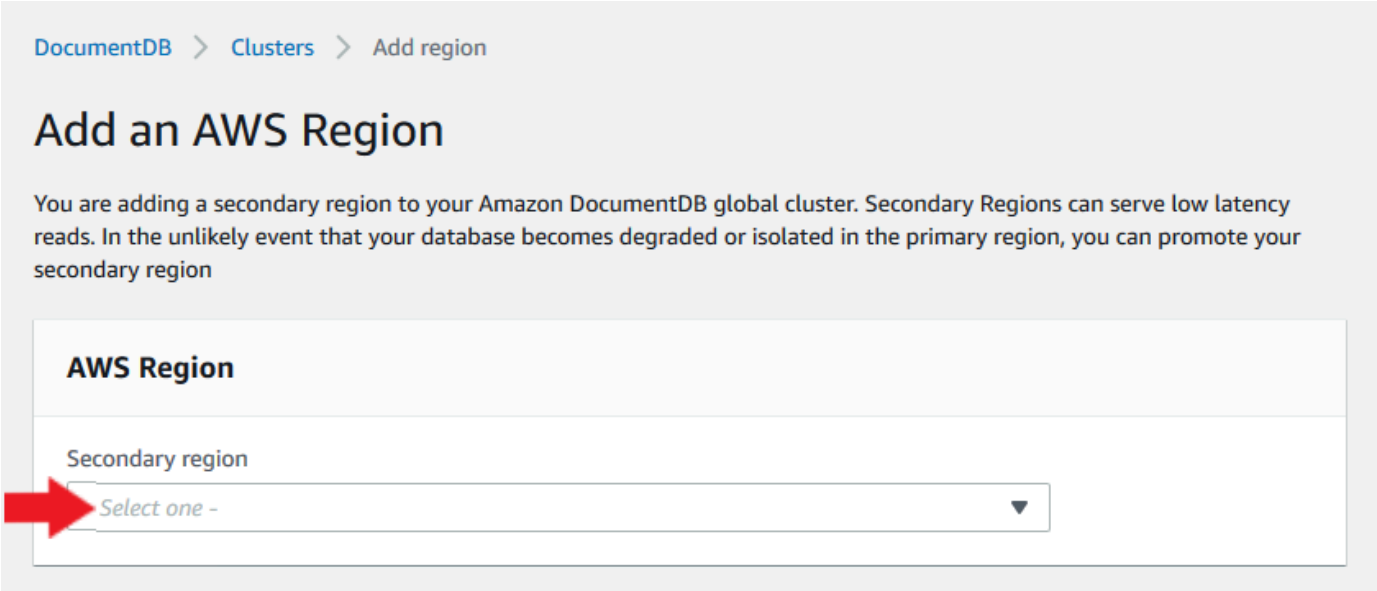
3. Choisissez le cluster auquel vous souhaitez ajouter un cluster secondaire. Assurez-vous que le cluster l'est`Available`.



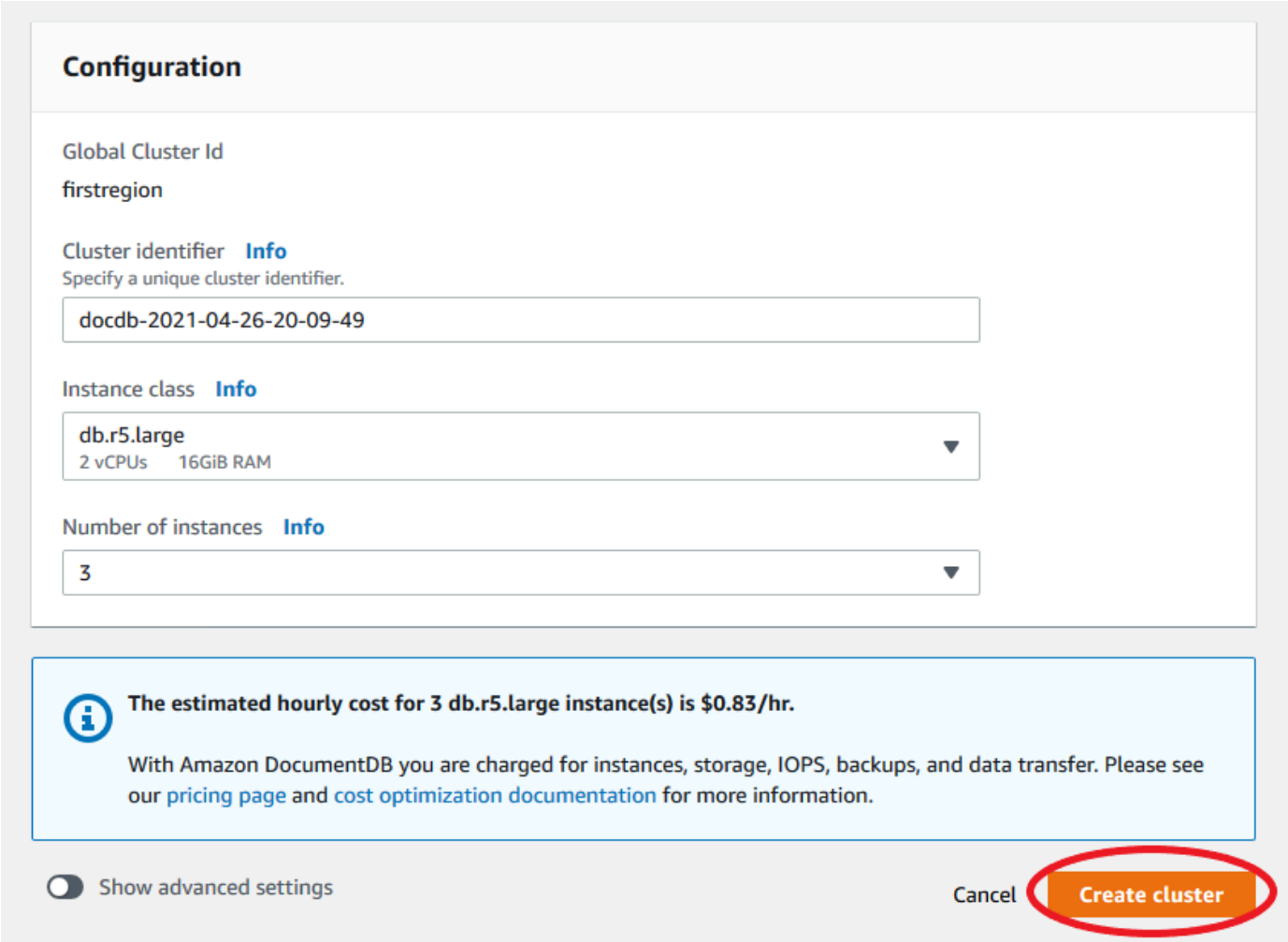
4. Sélectionnez le menu déroulant Actions, puis choisissez Ajouter une région.



5. Sur la page Ajouter une région, choisissez la région secondaire. Notez que vous ne pouvez pas choisir une région qui possède déjà un cluster secondaire pour le même cluster mondial. De plus, il ne peut pas s'agir de la même région que le cluster principal. S'il s'agit de la première région que vous ajoutez, vous devrez également spécifier un identifiant de cluster global de votre choix.



6. Complétez les champs restants pour le cluster secondaire dans la nouvelle région, puis sélectionnez Créer un cluster. Une fois que vous avez terminé d'ajouter la région, vous pouvez la voir dans la liste des clusters duAWS Management Console.



Utilisation de la AWS CLI

- Utilisez la commande `create-db-cluster` CLI avec le nom (`--global-cluster-identifier`) de votre cluster global. Pour les autres paramètres, procédez comme suit :
  - Pour`--region`, choisissez une région différente de Région AWS celle de votre région principale.
  - Choisissez des valeurs spécifiques pour les paramètres `--engine` et `--engine-version`.
  - Pour un cluster chiffré, spécifiez votre Région AWS principale comme `--source-region` pour le chiffrement.

L'exemple suivant crée un nouveau cluster Amazon DocumentDB et l'attache au cluster global en tant que cluster secondaire en lecture seule. Dans la dernière étape, l'instance est ajoutée au nouveau cluster.

Dans les exemples suivants, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations.

Pour Linux, macOS ou Unix :

```
aws docdb --region secondary-region-id \
  create-db-cluster \
    --db-cluster-identifier cluster-id \
    --global-cluster-identifier global-cluster-id \
    --engine-version version \
    --engine docdb

aws docdb --region secondary-region-id \
  create-db-instance \
    --db-cluster-identifier cluster-id \
    --global-cluster-identifier global-cluster-id \
    --engine-version version \
    --engine docdb
```

Pour Windows :

```
aws docdb --region secondary-region-id ^
  create-db-cluster ^
    --db-cluster-identifier cluster-id ^
    --global-cluster-identifier global-cluster-id ^
    --engine-version version ^
    --engine docdb

aws docdb --region secondary-region-id ^
  create-db-instance ^
    --db-cluster-identifier cluster-id ^
    --global-cluster-identifier global-cluster-id ^
    --engine-version version ^
    --engine docdb
```

Utilisation d'un instantané pour votre cluster global Amazon DocumentDB

Vous pouvez restaurer un instantané d'un cluster Amazon DocumentDB afin de l'utiliser comme point de départ pour votre cluster global. Pour ce faire, vous devez restaurer le snapshot et créer un

nouveau cluster. Il servira de cluster principal de votre cluster mondial. Vous pouvez ensuite ajouter une autre région au cluster restauré, le convertissant ainsi en cluster global.

Gestion d'un cluster global Amazon DocumentDB

Vous effectuez la plupart des opérations de gestion sur les clusters individuels qui constituent un cluster global. Lorsque vous sélectionnez Grouper les ressources associées sur la page Clusters de la console, le cluster principal et les clusters secondaires sont regroupés sous le cluster global associé.

L'onglet Configuration d'un cluster global indique l' Régions AWS endroit où les clusters sont exécutés, la version et l'identifiant du cluster global.

Rubriques

Modification d'un cluster global Amazon DocumentDB

La page Clusters AWS Management Console répertorie tous vos clusters globaux, en indiquant le cluster principal et les clusters secondaires pour chacun d'eux. Le cluster global possède ses propres paramètres de configuration. Plus précisément, des régions sont associées à ses clusters principaux et secondaires.

Lorsque vous apportez des modifications au cluster global, vous avez la possibilité d'annuler les modifications.

Lorsque vous choisissez Continuer, vous confirmez les modifications.

Modification des paramètres d'un cluster global Amazon DocumentDB

Vous pouvez configurer les groupes de paramètres de cluster indépendamment pour chaque cluster au sein du cluster global. La plupart des paramètres fonctionnent de la même manière que pour les autres types de clusters Amazon DocumentDB. Nous vous recommandons de maintenir la cohérence

des paramètres entre tous les clusters d'une base de données globale. Vous pourrez ainsi éviter les changements de comportement inattendus si vous choisissez un cluster secondaire en tant que cluster principal.

Par exemple, utilisez les mêmes paramètres pour les fuseaux horaires et les jeux de caractères afin d'éviter tout écart de comportement si un autre cluster devient le cluster principal.

Supprimer un cluster d'un cluster global Amazon DocumentDB

Il existe plusieurs situations dans lesquelles vous souhaiterez peut-être supprimer des clusters de votre cluster global. Par exemple, vous souhaiterez peut-être supprimer un cluster d'un cluster global si le cluster principal est dégradé ou isolé. Il devient alors un cluster provisionné autonome qui peut être utilisé pour créer un nouveau cluster global. Pour en savoir plus, consultez la section Restauration manuelle d'un cluster global suite à une panne imprévue.

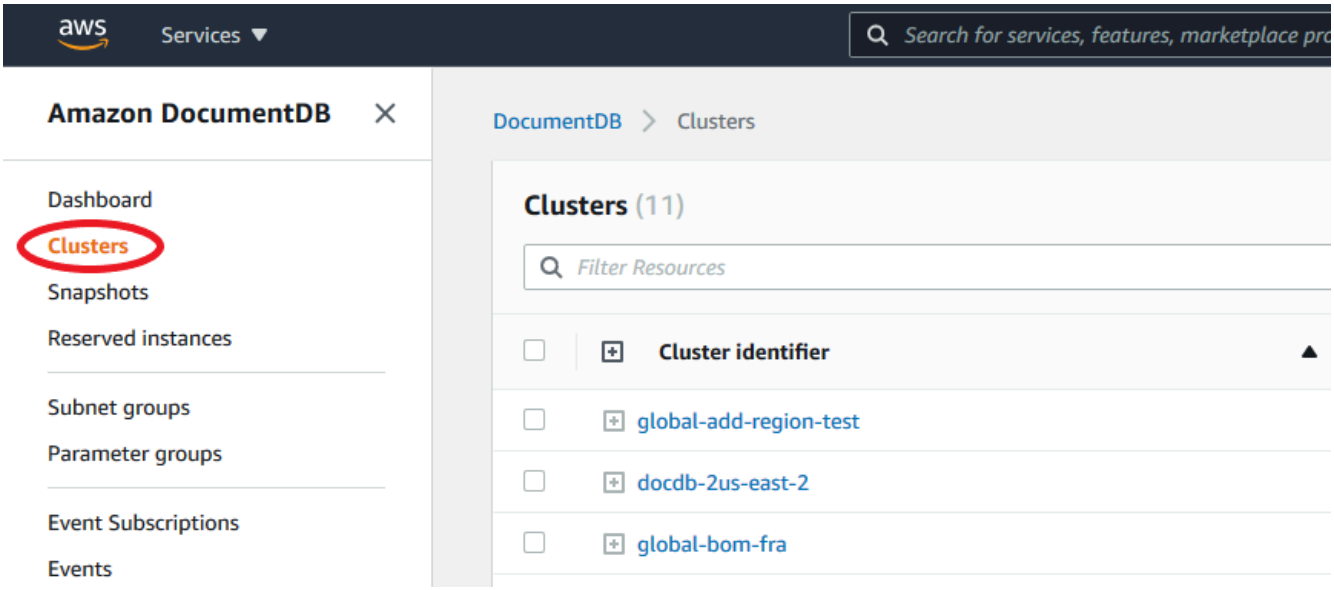
Vous souhaiterez peut-être également supprimer des clusters car vous souhaitez supprimer un cluster global dont vous n'avez plus besoin. Vous ne pouvez pas supprimer le cluster global avant d'avoir détaché tous les clusters associés, en laissant le cluster principal en dernier. Pour plus d'informations, consultez Supprimer un cluster global Amazon DocumentDB.

> **Note**
>
> Lorsqu'un cluster est détaché du cluster global, il n'est plus synchronisé avec le cluster principal. Il devient un cluster provisionné autonome doté de fonctionnalités complètes de lecture/écriture. En outre, il n'est plus visible dans la console Amazon DocumentDB. Il n'est visible que lorsque vous sélectionnez la région de la console dans laquelle se trouvait le cluster.

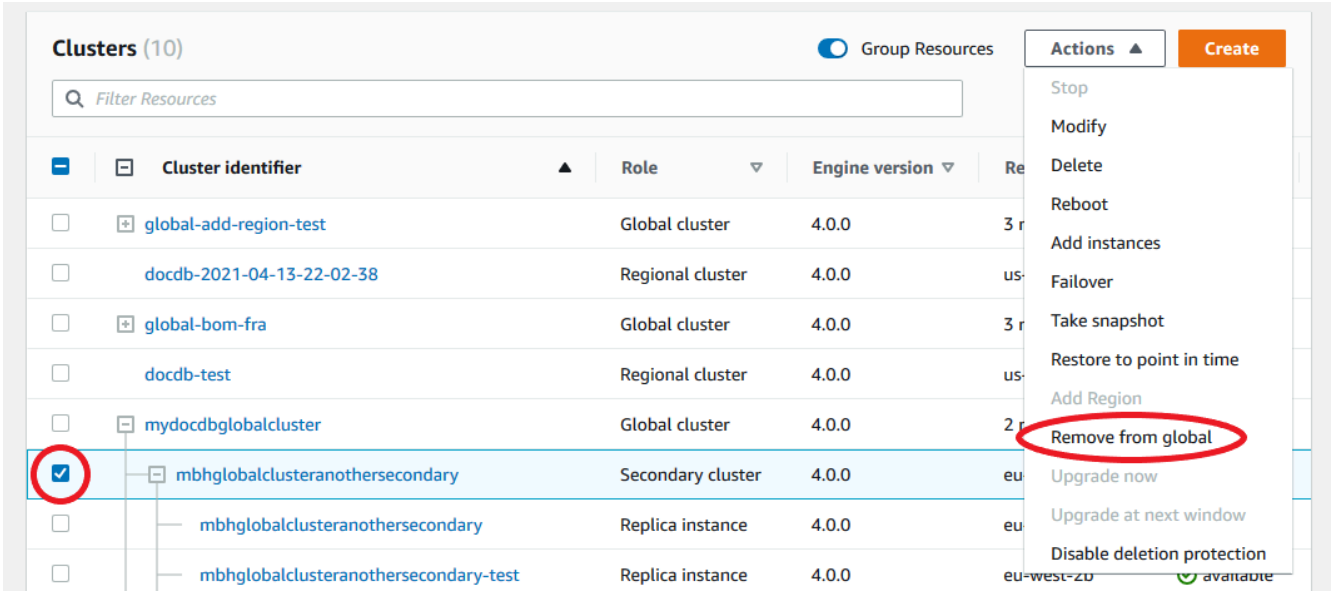
Vous pouvez supprimer des clusters de votre cluster global à l'aide de l'API AWS Management Console, de AWS CLI, ou de l'API RDS.

Using the AWS Management Console

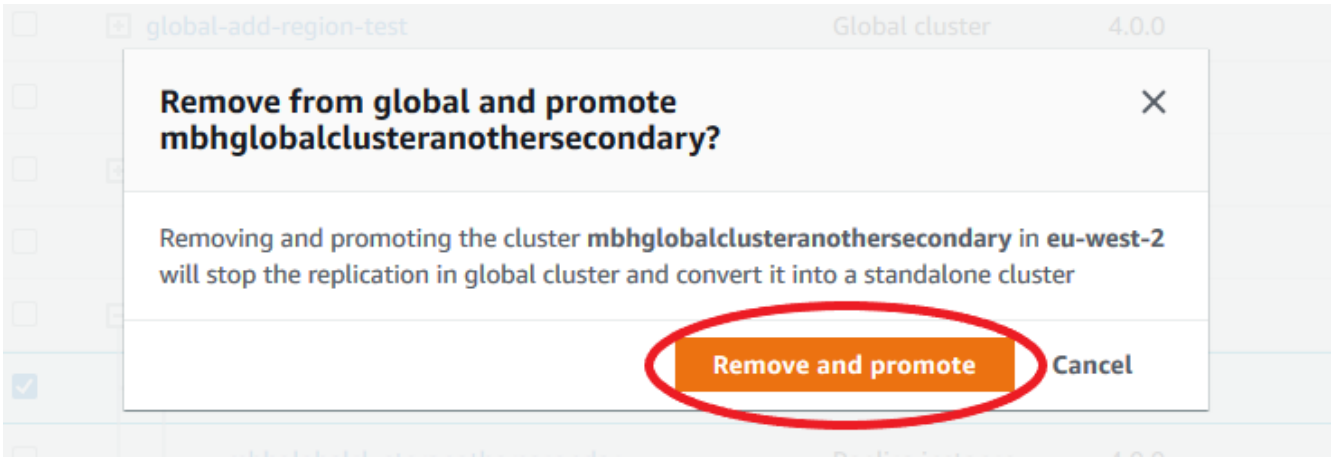
1. Connectez-vous à la console Amazon DocumentDB AWS Management Console et accédez à celle-ci.
2. Choisissez Clusters dans la barre de navigation de gauche.



3. Développez le cluster global afin de voir tous les clusters secondaires. Sélectionnez les clusters secondaires que vous souhaitez supprimer. Choisissez Actions, puis dans le menu déroulant, choisissez Supprimer du global.



4. Une invite s'affichera, vous demandant de confirmer que vous souhaitez détacher le secondaire du cluster global. Choisissez Supprimer et promouvoir pour supprimer le cluster du cluster global.



Désormais, ce cluster ne sert plus de cluster secondaire et n'est plus synchronisé avec le cluster principal. Il s'agit d'un cluster autonome doté d'une capacité de lecture/écriture complète.

Après avoir dissocié ou supprimé les clusters secondaires, vous pouvez procéder de la même façon pour dissocier le cluster principal. Vous ne pouvez pas détacher ou supprimer le cluster principal du cluster global tant que vous n'avez pas supprimé tous les clusters secondaires. Le cluster global peut rester dans la liste des clusters, sans aucune région ni zone de disponibilité. Vous pouvez le supprimer si vous ne souhaitez plus utiliser ce cluster global.

Using the AWS CLI

Pour supprimer un cluster d'un cluster global, exécutez la commande `remove-from-global-cluster` CLI avec les paramètres suivants :

- `--global-cluster-identifier`— Le nom (identifiant) de votre cluster mondial.
- `--db-cluster-identifier`— Le nom de chaque cluster à supprimer du cluster global.

Les exemples suivants suppriment d'abord un cluster secondaire, puis le cluster principal d'un cluster global.

Pour Linux, macOS ou Unix :

```
aws docdb --region secondary_region \
  remove-from-global-cluster \
    --db-cluster-identifier secondary_cluster_ARN \
    --global-cluster-identifier global_cluster_id

aws docdb --region primary_region \
```

```
  remove-from-global-cluster \
    --db-cluster-identifier primary_cluster_ARN \
    --global-cluster-identifier global_cluster_id
```

Répétez la `remove-from-global-cluster --db-cluster-identifier secondary_cluster_ARN` commande pour chaque région secondaire de votre cluster global.

Pour Windows :

```
aws docdb --region secondary_region ^
  remove-from-global-cluster ^
    --db-cluster-identifier secondary_cluster_ARN ^
    --global-cluster-identifier global_cluster_id

aws docdb --region primary_region ^
  remove-from-global-cluster ^
    --db-cluster-identifier primary_cluster_ARN ^
    --global-cluster-identifier global_cluster_id
```

Répétez la `remove-from-global-cluster --db-cluster-identifier secondary_cluster_ARN` commande pour chaque région secondaire de votre cluster global.

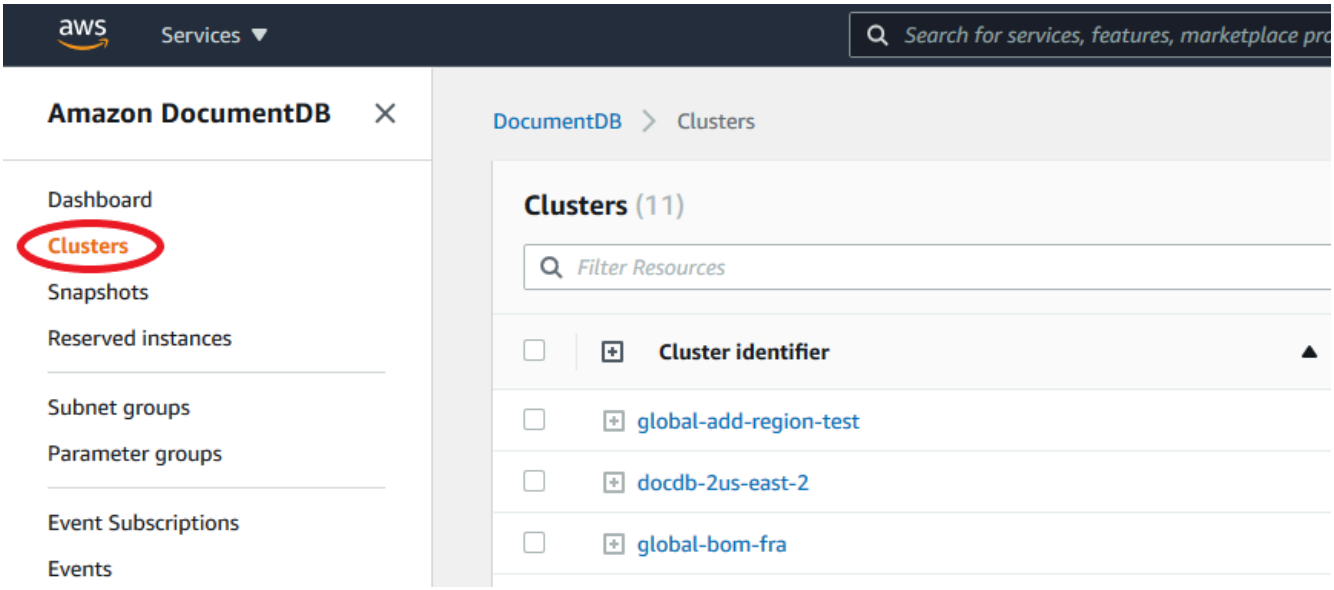
Supprimer un cluster d'un cluster global Amazon DocumentDB

Pour supprimer un cluster global, procédez comme suit :

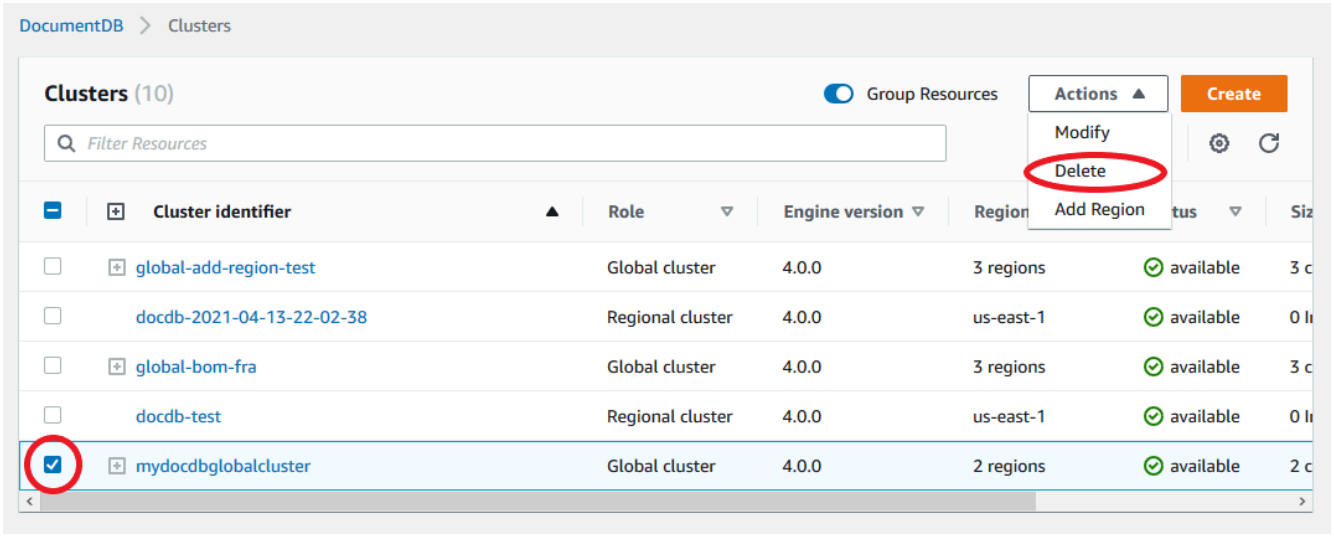
- Supprimez tous les clusters secondaires du cluster global. Chaque cluster devient un cluster autonome. Reportez-vous à la section précédente, Suppression de clusters globaux.
- Dans chaque cluster autonome, supprimez toutes les répliques.
- Supprimez le cluster principal du cluster global. Cela devient un cluster autonome.
- Dans le cluster principal, supprimez d'abord toutes les répliques, puis supprimez l'instance principale. La suppression de l'instance principale du nouveau cluster autonome entraîne généralement la suppression du cluster et du cluster global.

Using the AWS Management Console

1. Connectez-vous à la console Amazon DocumentDB AWS Management Console et accédez à celle-ci.
2. Choisissez Clusters et recherchez le cluster global que vous souhaitez supprimer.



3. Une fois votre cluster global sélectionné, choisissez Supprimer dans le menu Actions.



Vérifiez que tous les clusters sont supprimés du cluster global. Le cluster mondial ne doit afficher aucune région ni zone de disponibilité et une taille de zéro cluster. Si le cluster global contient des clusters, vous ne pouvez pas encore le supprimer. Vous devez d'abord suivre les instructions de l'étape précédente, Suppression des clusters globaux.

Using the AWS CLI

Pour supprimer un cluster global, exécutez la commande `delete-global-cluster` CLI avec le nom Région AWS et l'identifiant du cluster global, comme indiqué dans l'exemple suivant.

Pour Linux, macOS ou Unix :

```
aws docdb --region primary_region delete-global-cluster \
   --global-cluster-identifier global_cluster_id
```

Pour Windows :

```
aws docdb --region primary_region delete-global-cluster ^
   --global-cluster-identifier global_cluster_id
```

Création d'un cluster Amazon DocumentDB sans tête dans une région secondaire

Bien qu'un cluster global Amazon DocumentDB nécessite au moins un cluster secondaire différent Région AWS du cluster principal, vous pouvez utiliser une configuration headless pour le cluster secondaire. Un cluster Amazon DocumentDB secondaire sans tête est un cluster sans instance. Ce type de configuration permet de réduire les dépenses d'un cluster global. Dans un cluster Amazon DocumentDB, le calcul et le stockage sont découplés. Sans l'instance, le calcul ne vous est pas facturé, uniquement le stockage. S'il est correctement configuré, le volume de stockage d'un périphérique secondaire sans tête est synchronisé avec le cluster principal.

Vous ajoutez le cluster secondaire comme vous le faites habituellement lors de la création d'un cluster global Amazon DocumentDB. Toutefois, une fois que le cluster principal a commencé la réplication vers le cluster secondaire, vous supprimez l'instance en lecture seule du cluster secondaire. Ce cluster secondaire est désormais considéré comme « sans tête » car il ne possède plus d'instance. Cependant, le volume de stockage reste synchronisé avec le cluster Amazon DocumentDB principal.

> **Important**
>
> Nous recommandons uniquement les clusters headless aux clients qui peuvent tolérer des pannes régionales pendant plus de 15 minutes. En effet, la restauration après une défaillance régionale avec un cluster secondaire sans tête obligera l'utilisateur à créer une nouvelle instance après le basculement. La disponibilité d'une nouvelle instance peut prendre environ 10 à 15 minutes.

Comment ajouter un cluster secondaire sans tête à votre cluster mondial

1. Connectez-vous à la console Amazon DocumentDB AWS Management Console et ouvrez-la.

2. Choisissez Clusters dans la barre de navigation de gauche.
3. Choisissez le cluster global qui a besoin d'un cluster secondaire. Assurez-vous que le cluster principal est`Available`.
4. Pour Actions, choisissez Add region (Ajouter une région).
5. Sur la page Ajouter une région, choisissez la région secondaire.

> **Note**
>
> Vous ne pouvez pas choisir une région qui possède déjà un cluster secondaire pour le même cluster mondial. De plus, il ne peut pas s'agir de la même région que le cluster principal.

6. Complétez les champs restants pour le cluster secondaire dans la nouvelle région. Il s'agit des mêmes options de configuration que pour n'importe quelle instance de cluster.
7. Ajoutez une région. Une fois que vous avez terminé d'ajouter la région à votre cluster mondial, vous la verrez `Clusters` dans la liste du AWS Management Console.
8. Vérifiez l'état du cluster secondaire et de son instance de lecteur avant de continuer, en utilisant le AWS Management Console ou le AWS CLI. Voici un exemple de commande si vous utilisez AWS CLI :

```
$ aws docdb describe-db-clusters --db-cluster-identifier secondary-cluster-id --
query '*[].[Status]' --output text
```

   Plusieurs minutes peuvent être nécessaires pour que le statut d'un cluster secondaire récemment ajouté passe de « création » à « disponible ». Lorsque le cluster est disponible, vous pouvez supprimer l'instance de lecteur.
9. Sélectionnez l'instance de lecteur dans le cluster secondaire, puis choisissez Supprimer.
10. Après avoir supprimé l'instance de lecteur, le cluster secondaire fait toujours partie du cluster global. Aucune instance ne doit lui être associée.

> **Note**
>
> Vous pouvez utiliser ce cluster Amazon DocumentDB secondaire sans tête pour récupérer manuellement votre cluster global Amazon DocumentDB suite à une panne imprévue dans la région principale si une telle panne se produit.

Connectez-vous à un cluster global Amazon DocumentDB

La façon dont vous vous connectez à un cluster global varie selon que vous devez écrire dans le cluster ou lire depuis le cluster :

- Pour les demandes ou requêtes en lecture seule, vous vous connectez au point de terminaison du lecteur pour le cluster de votre. Région AWS
- Pour exécuter des instructions DML ou DDL, vous vous connectez au point de terminaison de cluster du cluster principal. Ce point de terminaison se trouve peut-être dans une autre application Région AWS que celle de votre application.

Lorsque vous visualisez un cluster global dans la console, vous pouvez voir tous les points de terminaison à usage général associés à tous ses clusters.

La façon dont vous vous connectez à un cluster global dépend de la nécessité d'écrire dans la base de données ou de lire des informations dans la base de données. Pour les opérations DDL, DML et de lecture que vous souhaitez effectuer depuis la région principale, vous devez vous connecter à votre cluster principal. Nous vous recommandons de vous connecter à votre cluster principal en utilisant le point de terminaison du cluster en mode Replica Set, avec une préférence de lecture de`secondaryPreferred=true`. Cela acheminera le trafic d'écriture vers l'instance d'écriture de votre cluster principal et le trafic de lecture vers l'instance de réplique de votre cluster principal.

Pour le trafic interrégional en lecture seule, vous devez vous connecter à l'un de vos clusters secondaires. Nous vous recommandons de vous connecter à votre cluster secondaire en utilisant le point de terminaison du cluster en mode Replica Set. Comme toutes les instances sont des instances de réplication en lecture seule, il n'est pas nécessaire de spécifier de préférence de lecture. Pour minimiser la latence, choisissez le point de terminaison du lecteur situé dans votre région ou dans la région la plus proche de chez vous.

Surveillance des clusters globaux Amazon DocumentDB

Amazon DocumentDB (compatible avec MongoDB) s'intègre CloudWatch afin que vous puissiez collecter et analyser les métriques opérationnelles de vos clusters. Vous pouvez surveiller ces métriques à l'aide de la CloudWatch console, de la console Amazon DocumentDB, du AWS Command Line Interface (AWS CLI) ou de l' CloudWatch API.

Pour surveiller un cluster global, utilisez les CloudWatch mesures suivantes.

Métrique	Description
`GlobalClusterReplicatedWriteIO`	Nombre moyen d'opérations d'E/S d'écritur e facturées répliquées depuis le volume de cluster du volume principal Région AWS vers le volume de cluster du volume secondair e Région AWS, indiqué à intervalles de 5 minutes. Le nombre de réplications dans `ReplicatedWriteIOs` chaque région secondaire est le même que le nombre de réplications `VolumeWriteIOPs` effectuées dans la région par la région principale.
`GlobalClusterDataTransferBytes`	La quantité de données transférée du cluster principal Région AWS vers le cluster secondair e est Région AWS mesurée en octets.
`GlobalClusterReplicationLag`	Le délai, en millisecondes, lors de la réplicati on des événements de changement du cluster principal Région AWS vers celui d'un cluster secondaire Région AWS

Pour plus d'informations sur la façon de consulter ces statistiques, consultez la section Affichage CloudWatch des données.

Reprise après sinistre et clusters globaux Amazon DocumentDB

En utilisant un cluster mondial, vous pouvez vous remettre rapidement après des catastrophes telles que des défaillances régionales. La reprise après sinistre est généralement mesurée à l'aide de valeurs RTO et RPO.

- Objectif de délai de reprise (RTO) : temps nécessaire à un système pour revenir à un état de fonctionnement normal après un sinistre. En d'autres termes, le RTO mesure les temps d'arrêt. Pour un cluster mondial, le RTO peut être de l'ordre de quelques minutes.
- Objectif de point de reprise (RPO) : quantité de données pouvant être perdues (mesurée dans le temps). Pour un cluster mondial, le RPO est généralement mesuré en secondes.

- Pour vous remettre d'une panne imprévue, vous pouvez effectuer un basculement entre régions vers l'un des centres secondaires de votre cluster mondial. Lorsque votre cluster global comporte plusieurs régions secondaires, assurez-vous de détacher toutes les régions secondaires en cas de panne de Région AWS la principale. Ensuite, vous promouvez l'une de ces régions secondaires en tant que nouvelle région principale Région AWS. Enfin, vous créez de nouveaux clusters dans chacune des autres régions secondaires et vous attachez ces clusters à votre cluster global.
- Lorsque vous promouvez un cluster secondaire comme cluster principal, vous devez également mettre à jour les points de terminaison utilisés par vos applications pour se connecter au cluster global. Pour obtenir un nouveau point de terminaison de rédacteur à partir d'un cluster nouvellement promu, vous pouvez convertir un ancien point de terminaison de lecteur en supprimant `-ro` de la chaîne de point de terminaison. Par exemple, si un ancien point de terminaison de lecteur est `global-16rr-test-cluster-1.cluster-ro-12345678901.us-west-2.docdb.amazonaws.com`, le nouveau point de terminaison de rédacteur promu est `global-16rr-test-cluster-1.cluster-cps2igpwyrwa.us-west-2.rds.amazonaws.com`.

Basculement sur incident pour les clusters mondiaux Amazon DocumentDB

Si un cluster entier dans un cluster Région AWS devient indisponible, vous pouvez promouvoir un autre cluster du cluster global pour qu'il dispose d'une capacité de lecture/écriture.

Vous pouvez activer manuellement le mécanisme de basculement s'il Région AWS est préférable de choisir un cluster situé dans un autre cluster comme cluster principal. Par exemple, vous pouvez accroître la capacité de l'un des clusters secondaires, puis le promouvoir comme cluster principal. L'équilibre des activités entre eux Régions AWS peut également changer, de sorte que le fait de passer du cluster principal à un autre Région AWS peut réduire le temps de latence pour les opérations d'écriture.

La procédure suivante décrit la procédure à suivre pour promouvoir l'un des clusters secondaires d'un cluster global DocumentDB.

Pour promouvoir un cluster secondaire :

1. Arrêtez d'émettre des instructions DML et d'autres opérations d'écriture sur le cluster principal en cas Région AWS de panne.
2. Identifiez un cluster à partir d'un cluster secondaire Région AWS à utiliser comme nouveau cluster principal. Si vous en avez deux (ou plus) Régions AWS dans votre cluster global, choisissez le cluster secondaire qui présente le moins de temps de latence.

3. Détachez le cluster secondaire que vous avez choisi du cluster global.

   La suppression d'un cluster secondaire d'un cluster global arrête immédiatement la réplication du cluster principal vers ce cluster secondaire et en fait un cluster de cluster autonome provisionné doté de fonctionnalités de lecture/écriture complètes. Tout autre cluster secondaire associé au cluster principal dans la région touchée par la panne est toujours disponible et peut accepter les appels de votre application. Ils consomment également des ressources. Puisque vous recréez le cluster global, pour éviter les problèmes liés au split brain et à d'autres problèmes, supprimez les autres clusters secondaires avant de créer le nouveau cluster global en suivant les étapes ci-dessous.

   Afin d'obtenir les étapes détaillées du détachement, consultez Supprimer un cluster d'un cluster global Amazon DocumentDB.

4. Reconfigurez votre application pour envoyer toutes les opérations d'écriture à ce cluster désormais autonome à l'aide de son nouveau point de terminaison. Si vous avez accepté les noms fournis lors de la création du cluster global, vous pouvez modifier le point de terminaison en supprimant le -ro de la chaîne de point de terminaison du cluster dans votre application.

   Par exemple, le point de terminaison du cluster secondaire `my-global.cluster-ro-aaaaaabbbbbb.us-west-1.docdb.amazonaws.com` devient `my-global.cluster-aaaaaabbbbbb.us-west-1.docdb.amazonaws.com` lorsque ce cluster est détaché du cluster global.

   Ce cluster devient le cluster principal d'un nouveau cluster mondial lorsque vous commencez à y ajouter des régions, à l'étape suivante.

5. Ajoutez un Région AWS au cluster. Lorsque vous effectuez cette opération, le processus de réplication du cluster primaire vers le cluster secondaire commence.

6. Ajoutez-en Régions AWS d'autres si nécessaire pour recréer la topologie requise pour prendre en charge votre application. Assurez-vous que les écritures de l'application sont envoyées au cluster approprié avant, pendant et après de telles modifications, afin d'éviter les incohérences de données entre les clusters du cluster global (problèmes liés au split brain).

7. Lorsque la panne est résolue et que vous êtes prêt à réattribuer votre cluster d'origine Région AWS comme cluster principal, effectuez les mêmes étapes dans le sens inverse.

8. Supprimez l'un des clusters secondaires du cluster global. Cela lui permettra de desservir le trafic de lecture/écriture.

9. Redirigez tout le trafic d'écriture vers le cluster principal dans l'original Région AWS.

10. Ajoutez un Région AWS pour configurer un ou plusieurs clusters secondaires de la même manière Région AWS que précédemment.

Les clusters globaux Amazon DocumentDB peuvent être gérés à l'aide de AWS kits SDK, ce qui vous permet de créer des solutions pour automatiser le processus de basculement des clusters mondiaux pour les cas d'utilisation liés à la reprise après sinistre et à la planification de la continuité des activités. L'une de ces solutions est mise à la disposition de nos clients sous licence Apache 2.0 et est accessible depuis notre référentiel d'outils ici. Cette solution utilise Amazon Route53 pour la gestion des terminaux et fournit des fonctions AWS Lambda qui peuvent être déclenchées en fonction d'événements appropriés.

Gestion des clusters Amazon DocumentDB

Pour gérer un cluster Amazon DocumentDB, vous devez disposer d'une politique IAM avec les autorisations appropriées du plan de contrôle Amazon DocumentDB. Ces autorisations vous permettent de créer, modifier et supprimer des instances et des clusters. La `AmazonDocDBFullAccess` politique fournit toutes les autorisations requises pour administrer un cluster Amazon DocumentDB.

Les rubriques suivantes expliquent comment effectuer différentes tâches lorsque vous travaillez avec des clusters Amazon DocumentDB, notamment la création, la suppression, la modification, la connexion et l'affichage de clusters.

Rubriques

- Comprendre les clusters
- Paramètres du cluster Amazon DocumentDB
- Configurations de stockage en cluster Amazon DocumentDB
- Déterminer le statut d'un cluster
- Cycle de vie du cluster Amazon DocumentDB
- Dimensionnement des clusters Amazon DocumentDB
- Clonage d'un volume pour un cluster Amazon DocumentDB
- Comprendre la tolérance aux pannes des clusters Amazon DocumentDB

Comprendre les clusters

Amazon DocumentDB sépare le calcul du stockage, et décharge la réplication et la sauvegarde des données vers le volume du cluster. Un volume de cluster fournit une couche de stockage durable, fiable et hautement disponible qui réplique les données de six façons dans trois zones de disponibilité. Les réplicas permettent une haute disponibilité des données et une mise à l'échelle en lecture. Chaque cluster peut redimensionner jusqu'à 15 répliques.

Nom	Description	Opérations d'API (Verbes)

Cluster	Il se compose d'une ou de plusieurs instances, et d'un volume de stockage de clusters qui gère les données pour ces instances.	`create-db-cluster`
		`delete-db-cluster`
		`describe-db-clusters`
		`modify-db-cluster`

Instance	La lecture et l'écriture de données sur le volume de stockage du cluster se font via des instances. Il existe deux types d'instances dans un cluster : une instance principale et un réplica. Un cluster possède toujours une instance principale et peut comporter de 0 à 15 répliques.	`create-db-instance`
		`delete-db-instance`
		`describe-db-instances`
		`modify-db-instance`
		`describe-orderable-db-instance-options`
		`reboot-db-instance`

Volume de cluster	un volume virtuel de stockage de base de données qui couvre trois zones de disponibilité, chacune d'entre elles ayant deux copies des données de cluster.	N/A

Nom	Description	Opérations d'API (Verbes)
Instance principale	Prend en charge les opérations de lecture et d'écriture, et effectue toutes les modifications de données du volume de cluster. Chaque cluster possède une seule instance principale.	N/A
Instance de réplica	Prend uniquement en charge les opérations de lecture. Chaque cluster Amazon DocumentDB peut comporter jusqu'à 15 instances de réplication en plus de l'instance principale. Plusieurs réplicas distribuent la charge de travail en lecture. En plaçant les réplicas dans des zones de disponibilité distinctes, vous pouvez aussi accroître la disponibilité de la base de données.	N/A
Point de terminaison de cluster	Point de terminaison pour un cluster Amazon DocumentDB qui se connecte à l'instance principale actuelle du cluster. Chaque cluster Amazon DocumentDB possède un point de terminaison de cluster et une instance principale.	N/A

Nom	Description	Opérations d'API (Verbes)
Point de terminaison du lecteur	Point de terminaison pour un cluster Amazon DocumentDB qui se connecte à l'une des répliques disponibles pour ce cluster. Chaque cluster Amazon DocumentDB possède un point de terminaison lecteur. S'il existe plusieurs répliques, le point de terminaison du lecteur dirige chaque demande de connexion vers l'une des répliques Amazon DocumentDB.	N/A
Point de terminaison d'instance	Point de terminaison pour une instance d'un cluster Amazon DocumentDB qui se connecte à une instance spécifique. Chaque instance d'un cluster, quel que soit le type d'instance, a son propre point de terminaison d'instance unique.	N/A

Paramètres du cluster Amazon DocumentDB

Lorsque vous créez ou modifier un cluster, il est important de comprendre quels paramètres sont immuables et lesquels sont modifiables, après la création du cluster. Le tableau suivant répertorie tous les paramètres qui sont spécifiques à un cluster. Comme indiqué dans le tableau, certains sont modifiables, d'autres ne le sont pas.

> **Note**
>
> Ces paramètres ne doivent pas être confondus avec les groupes de paramètres du cluster Amazon DocumentDB et leurs paramètres. Pour plus d'informations sur les groupes de

paramètres de cluster, consultez Gestion des groupes de paramètres du cluster Amazon DocumentDB.

Paramètre	Adaptabilité	Remarques
DBClusterIdentifier	Oui	Contraintes d'affectation de noms : • La longueur est de [1 à 63] lettres, chiffres ou traits d'union. • Le premier caractère doit être une lettre. • Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs. • Doit être unique pour tous les clusters d'Amazon Amazon RDS, Amazon Neptune et Amazon DocumentDB par région Compte AWS.
Engine	Non	Doit indiquer `docdb`.
BackupRetentionPeriod	Oui	Doit être comprise entre 1 et 35 jours.
DBClusterParameterGroupName	Oui	Contraintes d'affectation de noms : • La longueur est de [1 à 255] caractères alphanumériques. • Le premier caractère doit être une lettre. • Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
DBSubnetGroupName	Non	Une fois qu'un cluster a été créé, vous ne pouvez pas modifier son sous-réseau.
EngineVersion	Non	La valeur peut être `5.0.0` (par défaut)`4.0.0`, ou`3.6.0`.

Paramètre	Adaptabilité	Remarques
KmsKeyId	Non	Si vous choisissez de chiffrer votre cluster, vous ne pouvez pas modifier la AWS KMS clé que vous avez utilisée pour chiffrer votre cluster.
MasterUsername	Non	Une fois qu'un cluster a été créé, vous ne pouvez pas modifier le `MasterUsername` . Contraintes d'affectation de noms : • La longueur est de [1 à 63] caractères alphanumériques. • Le premier caractère doit être une lettre. • Ne peut pas être un mot réservé du moteur de base de données.

MasterUserPassword	Oui	Contraintes :
		
		• La longueur est de [8 à 100] caractères ASCII imprimables.
		• Tous les caractères ASCII imprimables peuvent être utilisés, à l'exception des suivants :
		  • **/** (barre oblique)
		  • **"** (guillemets doubles)
		  • **@** (symbole arobase)

Port	Oui	Le numéro de port s'applique à toutes les instances dans le cluster.
PreferredBackupWindow	Oui	
PreferredMaintenanceWindow	Oui	

Paramètre	Adaptabilité	Remarques
`StorageEncrypted`	Non	Si vous choisissez de chiffrer votre cluster, il ne peut pas être non chiffré.
`StorageType`	Oui	Type de stockage pour le cluster de base de données : Standard (`standard`) ou I/O-Optim ized (`iopt1`). Par défaut : `standard` Ce paramètre peut être configuré avec `CreateDBCluster etModifyDBCluster .` Pour plus d'informations, consultez Configura tions de stockage en cluster Amazon DocumentDB.
`Tags`	Oui	
`VpcSecurityGroupIds`	Non	Une fois qu'un cluster a été créé, vous ne pouvez pas modifier le VPC dans lequel réside le cluster.

Configurations de stockage en cluster Amazon DocumentDB

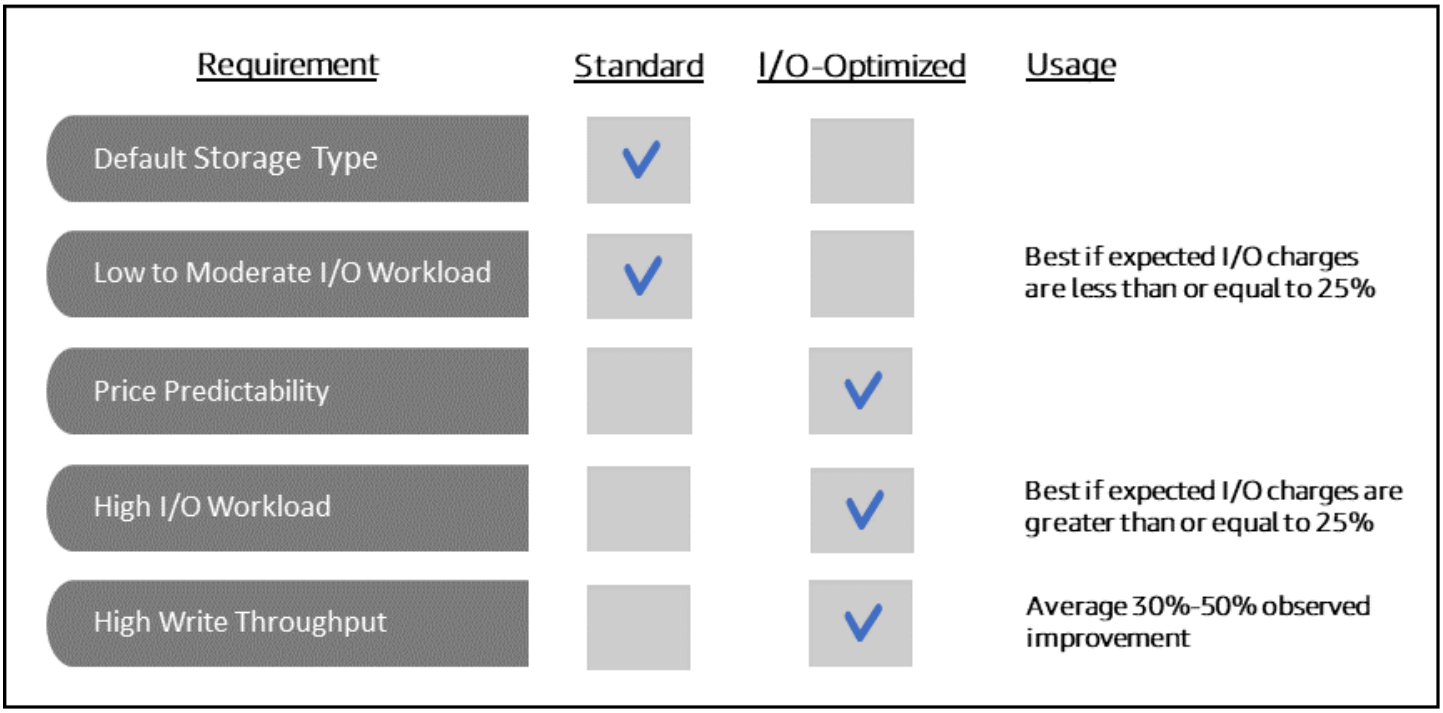
À partir d'Amazon DocumentDB 5.0, les clusters basés sur des instances prennent en charge deux types de configurations de stockage :

- Stockage standard Amazon DocumentDB : conçu pour les clients ayant une consommation d'E/S faible à modérée. Si vous pensez que vos coûts d'E/S seront inférieurs à 25 % de l'ensemble de votre cluster Amazon DocumentDB, cette option peut être idéale pour vous. Avec la configuration de stockage standard d'Amazon DocumentDB, vous êtes facturé sur la base des pay-per-request E/S, en plus des frais d'instance et de stockage. Cela signifie que votre facturation peut varier d'un cycle à l'autre en fonction de l'utilisation. La configuration est adaptée pour répondre aux demandes d'E/S fluctuantes de votre application.
- Stockage optimisé pour les E/S Amazon DocumentDB : conçu pour les clients qui privilégient la prévisibilité des prix ou qui ont des applications gourmandes en E/S. La configuration optimisée

pour les E/S offre des performances améliorées, un débit accru et une latence réduite pour les clients ayant des charges de travail intensives en E/S. Si vous pensez que vos coûts d'E/S dépasseront 25 % du coût total de votre cluster Amazon DocumentDB, cette option offre un meilleur rapport prix/performances. Grâce à la configuration de stockage optimisée pour les E/S d'Amazon DocumentDB, vous ne serez pas facturé en fonction des opérations d'E/S, ce qui garantit des coûts prévisibles à chaque cycle de facturation. La configuration stabilise les coûts tout en améliorant les performances.

Vous pouvez transférer vos clusters de bases de données existants une fois tous les 30 jours vers un stockage optimisé pour les E/S Amazon DocumentDB. Vous pouvez revenir au stockage standard Amazon DocumentDB à tout moment. La prochaine date de modification de la configuration de stockage pour qu'elle soit optimisée pour les E/S peut être suivie à l'aide de la `describe-db-clusters` commande à l'aide de AWS CLI ou via la AWS Management Console page de configuration du cluster.

Vous pouvez créer un nouveau cluster de bases de données incluant la configuration optimisée pour les E/S d'Amazon DocumentDB ou convertir vos clusters de bases de données existants en quelques clics AWS Management Console, en modifiant un seul paramètre dans le AWS Command Line Interface (AWS CLI) ou via des SDK.AWS Aucune interruption ou redémarrage des instances n'est nécessaire pendant ou après la modification de la configuration du stockage.

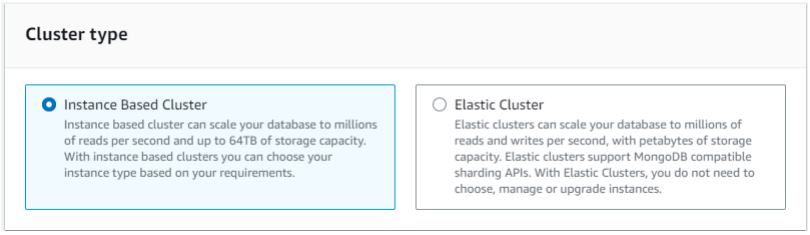


Création d'un cluster optimisé pour les E/S

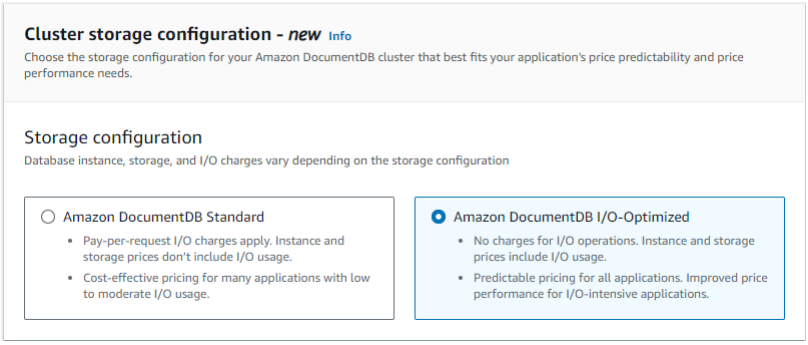
Using the AWS Management Console

Pour créer ou modifier un cluster optimisé pour les E/S à l'aide de : AWS Management Console

1. Sur la console de gestion Amazon DocumentDB, sous Clusters, choisissez Create ou sélectionnez le cluster et choisissez Actions, puis choisissez Modify.
2. Si vous créez un nouveau cluster, assurez-vous de choisir Instance Based Clusters dans la section Type de cluster (il s'agit de l'option par défaut).



3. Dans la section Configuration, sous Configuration du stockage en cluster, choisissez Amazon DocumentDB I/O Optimized.



4. Terminez la création ou la modification de votre cluster et choisissez Créer un cluster ou Modifier le cluster.

   Pour le processus complet de création d'un cluster, voirCréation d'un cluster et d'une instance principale à l'aide du AWS Management Console.

   Pour le processus complet de modification du cluster, voirModification d'un cluster Amazon DocumentDB.

Using the AWS CLI

Pour créer un cluster optimisé pour les E/S à l'aide de : AWS CLI

Dans les exemples suivants, remplacez chaque *user input placeholder* (espace réservé pour l'entrée utilisateur) avec vos propres informations.

Pour Linux, macOS ou Unix :

```
aws docdb create-db-cluster \
      --db-cluster-identifier sample-cluster \
      --engine docdb \
      --engine-version 5.0.0 \
      --storage-type iopt1 \
      --deletion-protection \
      --master-username username \
      --master-user-password password
```

Pour Windows :

```
aws docdb create-db-cluster ^
      --db-cluster-identifier sample-cluster ^
      --engine docdb ^
      --engine-version 5.0.0 ^
      --storage-type iopt1 ^
      --deletion-protection ^
      --master-username username ^
      --master-user-password password
```

Analyse des coûts pour déterminer la configuration du stockage

Avec Amazon DocumentDB, vous avez la possibilité de choisir votre configuration de stockage pour chaque cluster de base de données dont vous disposez. Afin de répartir correctement vos clusters entre les clusters standard et les clusters optimisés pour les E/S, vous pouvez suivre vos coûts Amazon DocumentDB par cluster. Pour ce faire, vous pouvez ajouter des balises aux clusters existants, activer le balisage de répartition des coûts dans votre AWS Billing and Cost Management tableau de bord et analyser vos coûts pour un cluster donné dans le AWS Cost Explorer Service. Pour plus d'informations sur l'analyse des coûts, consultez notre blog Utilisation des balises de répartition des coûts.

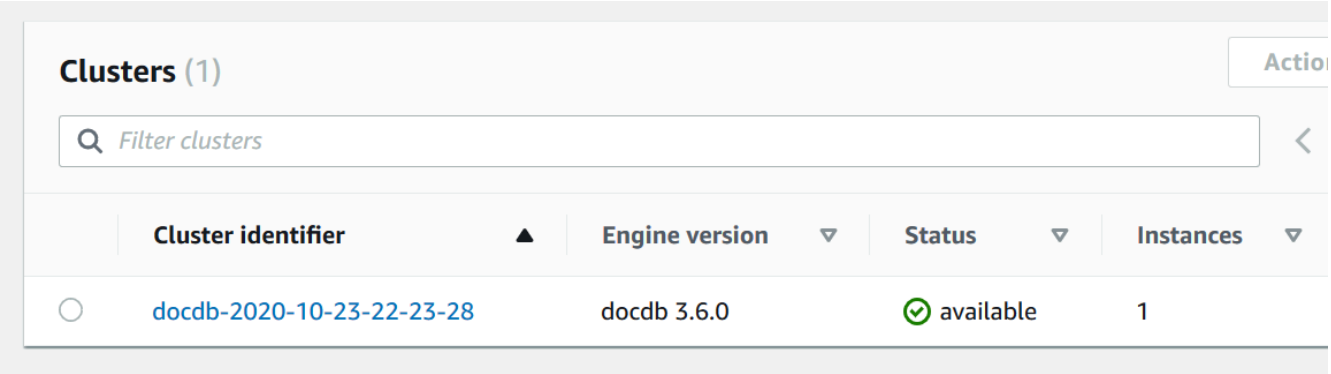
Déterminer le statut d'un cluster

Vous pouvez déterminer le statut d'un cluster à l'aide du AWS Management Console ou AWS CLI.

Using the AWS Management Console

Utilisez la procédure suivante pour connaître l'état de votre cluster Amazon DocumentDB à l'aide du AWS Management Console

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.
3. Dans la colonne Identifiant de cluster, recherchez le nom du cluster qui vous intéresse. Ensuite, pour trouver le statut du cluster, allez à la colonne Status (Statut), comme indiqué ci-dessous.



Using the AWS CLI

Utilisez l'`describe-db-clusters`opération pour connaître l'état de votre cluster Amazon DocumentDB à l'aide du. AWS CLI

Le code suivant trouve le statut du cluster `sample-cluster`.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster  \
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

Pour Windows :

```
aws docdb describe-db-clusters ^
    --db-cluster-identifier sample-cluster  ^
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

La sortie de cette opération ressemble à ceci (format JSON).

```
[
    [
        "sample-cluster",
        "available"
    ]
]
```

Cycle de vie du cluster Amazon DocumentDB

Le cycle de vie d'un cluster Amazon DocumentDB inclut la création, la description, la modification et la suppression du cluster. Cette section fournit des informations sur la façon de réaliser ces processus.

Rubriques

Création d'un cluster Amazon DocumentDB

Un cluster Amazon DocumentDB se compose d'instances et d'un volume de cluster qui représente les données du cluster. Le volume de cluster est répliqué six fois dans trois zones de disponibilité en tant que volume virtuel unique. Le cluster contient une instance principale et, éventuellement, jusqu'à 15 instances de réplica.

Les sections suivantes montrent comment créer un cluster Amazon DocumentDB à l'aide du AWS Management Console ou du. AWS CLI Vous pouvez ensuite ajouter des instances de réplica pour ce cluster. Lorsque vous utilisez la console pour créer votre cluster Amazon DocumentDB, une instance principale est automatiquement créée pour vous en même temps. Si vous utilisez le AWS CLI pour

créer votre cluster Amazon DocumentDB, une fois que l'état du cluster est disponible, vous devez créer l'instance principale pour ce cluster.

Prérequis

Les conditions requises pour créer un cluster Amazon DocumentDB sont les suivantes.

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

Pour vous inscrire à un Compte AWS

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

   Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique 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 en matière de sécurité consiste à attribuer un accès administratif à un utilisateur et à n'utiliser que l'utilisateur root pour effectuer les tâches nécessitant un accès utilisateur root.

Prérequis pour VPC

Vous ne pouvez créer un cluster Amazon DocumentDB que dans un Amazon Virtual Private Cloud (Amazon VPC). Votre Amazon VPC doit disposer d'au moins un sous-réseau dans chacune des deux zones de disponibilité au moins pour que vous puissiez l'utiliser avec un cluster Amazon DocumentDB. En répartissant vos instances de cluster entre les zones de disponibilité, vous vous assurez qu'elles sont disponibles dans votre cluster dans le cas peu probable d'une défaillance de la zone de disponibilité.

Conditions requises pour les sous-réseaux

Lorsque vous créez un cluster Amazon DocumentDB, vous devez choisir un VPC et le groupe de sous-réseaux correspondant au sein de ce VPC pour lancer votre cluster. Les sous-réseaux déterminent la zone de disponibilité et la plage d'adresses IP au sein de cette zone de disponibilité que vous souhaitez utiliser pour lancer une instance. Dans le cadre de cette présentation, nous utiliserons les termes sous-réseau et zone de disponibilité indifféremment. Un groupe de sous-

réseaux est un ensemble nommé de sous-réseaux (ou de zones de disponibilité). Un groupe de sous-réseaux vous permet de spécifier les zones de disponibilité que vous souhaitez utiliser pour lancer des instances Amazon DocumentDB. Par exemple, dans un cluster avec trois instances et pour une haute disponibilité, il est recommandé que chacune de ces instances soit mise en service dans des zones de disponibilité distinctes. Par conséquent, si une zone de disponibilité s'arrête, cela affecte une seule instance.

Les instances Amazon DocumentDB peuvent actuellement être mises en service dans un maximum de trois zones de disponibilité. Même si un groupe de sous-réseaux possède plus de trois sous-réseaux, vous ne pouvez utiliser que trois de ces sous-réseaux pour créer un cluster Amazon DocumentDB. Par conséquent, lors de la création d'un groupe de sous-réseaux, il est recommandé de choisir les trois sous-réseaux sur lesquels vous souhaitez déployer vos instances. Dans l'est des États-Unis (Virginie du Nord), votre groupe de sous-réseaux peut comporter six sous-réseaux (ou zones de disponibilité). Toutefois, lorsqu'un cluster Amazon DocumentDB est provisionné, Amazon DocumentDB choisit trois de ces zones de disponibilité qu'il utilise pour provisionner les instances.

Supposons, par exemple, que lorsque vous créez un cluster, Amazon DocumentDB choisisse les zones de disponibilité {1A, 1B et 1C}. Si vous essayez de créer une instance dans la zone de disponibilité {1D}, l'appel d'API échoue. Toutefois, si vous choisissez de créer une instance sans spécifier de zone de disponibilité particulière, Amazon DocumentDB choisit une zone de disponibilité en votre nom. Amazon DocumentDB utilise un algorithme pour équilibrer la charge des instances entre les zones de disponibilité afin de vous aider à atteindre une haute disponibilité. Par exemple, si trois instances sont allouées, elles le seront par défaut sur trois zones de disponibilité et non toutes dans une seule zone.

Recommandations :

- Sauf si vous avez une raison précise, créez toujours un groupe de sous-réseaux avec trois sous-réseaux. Cela permet de s'assurer que les clusters avec au moins trois instances sont en mesure de bénéficier d'une disponibilité plus importante, les instances étant allouées sur trois zones de disponibilité.
- Répartissez toujours les instances sur plusieurs zones de disponibilité pour obtenir une haute disponibilité. Ne placez jamais toutes les instances d'un cluster dans une seule zone de disponibilité.
- Des événements de basculement pouvant se produire à n'importe quel moment, vous ne devez pas supposer qu'une instance principale ou des instances de réplica sont toujours placées dans une zone de disponibilité spécifique.

Prérequis supplémentaires

Voici quelques prérequis supplémentaires pour créer un cluster Amazon DocumentDB :

- Si vous vous connectez à AWS l'aide d'informations d'identification AWS Identity and Access Management (IAM), votre compte IAM doit disposer de politiques IAM qui accordent les autorisations requises pour effectuer des opérations Amazon DocumentDB.

  Si vous utilisez un compte IAM pour accéder à la console Amazon DocumentDB, vous devez d'abord vous y connecter AWS Management Console avec votre compte IAM. Accédez ensuite à la console Amazon DocumentDB à l'adresse https://console.aws.amazon.com/docdb.
- Si vous voulez adapter les paramètres de configuration de votre cluster, vous devez spécifier un groupe de paramètres de cluster et un groupe de paramètres avec les valeurs de paramètre requises. Pour de plus amples informations sur la création ou la modification d'un groupe de paramètres de cluster ou d'un groupe de paramètres, veuillez consulter Gestion des groupes de paramètres du cluster Amazon DocumentDB.
- Vous devez déterminer le numéro de port TCP/IP que vous voulez spécifier pour votre cluster. Les pare-feux de certaines entreprises bloquent les connexions aux ports par défaut d'Amazon DocumentDB. Si le pare-feu de votre entreprise bloque le port par défaut, choisissez un autre port pour le cluster. Toutes les instances d'un cluster utilisent le même port.

Création d'un cluster et d'une instance principale à l'aide du AWS Management Console

Les procédures suivantes décrivent comment utiliser la console pour lancer un cluster Amazon DocumentDB avec une ou plusieurs instances.

Création d'un cluster : utilisation des paramètres par défaut

Pour créer un cluster avec des instances à l'aide des paramètres par défaut à l'aide du AWS Management Console

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Si vous souhaitez créer votre cluster dans une région Région AWS autre que l'est des États-Unis (Virginie du Nord), choisissez la région dans la liste située dans le coin supérieur droit de la console.
3. Dans le volet de navigation, sélectionnez Clusters, puis Create (Créer).

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu
> (≡)
> dans le coin supérieur gauche de la page.

4. Sur la page Créer un cluster Amazon DocumentDB, complétez le volet Configuration.

   a. Identifiant du cluster : acceptez le nom fourni par Amazon DocumentDB ou entrez un nom pour votre cluster, par exemple,. **`sample-cluster`**

      Contraintes d'attribution de nom relatives à un cluster :

      - La longueur est de [1 à 63] lettres, chiffres ou traits d'union.
      - Le premier caractère doit être une lettre.
      - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
      - Doit être unique pour tous les clusters d'Amazon RDS, Neptune et Amazon Compte AWS DocumentDB par région.

   b. Version du moteur : acceptez la version du moteur par défaut 4.0.0 ou choisissez éventuellement 3.6.0.

   c. Classe d'instance : acceptez la valeur par défaut `db.r5.large` ou choisissez la classe d'instance de votre choix dans la liste.

   d. Nombre d'instances : dans la liste, choisissez le nombre d'instances que vous souhaitez créer avec ce cluster. La première instance est l'instance principale, et toutes les autres instances sont des instances de réplica en lecture seule. Vous pouvez ajouter et supprimer des instances ultérieurement, si nécessaire. Par défaut, un cluster Amazon DocumentDB est lancé avec trois instances (une instance principale et deux répliques).

5. Complétez la section Configuration du stockage en cluster.

   Choisissez Amazon DocumentDB Standard (par défaut) ou Amazon DocumentDB I/O-Optimized. Pour plus d’informations, consultez Configurations de stockage en cluster Amazon DocumentDB.

6. Complétez le volet Authentication (Authentification).

a. Nom d'utilisateur —Entrez le nom de l'utilisateur principal. Pour vous connecter à votre cluster, vous devez utiliser le nom d'utilisateur principal.

   Contraintes de dénomination des utilisateurs principaux :

   - La longueur est de [1 à 63] caractères alphanumériques.
   - Le premier caractère doit être une lettre.
   - Ne peut pas être un mot réservé du moteur de base de données.

b. Mot de passe —Entrez un mot de passe pour l'utilisateur principal, puis confirmez-le. Pour vous connecter à votre cluster, vous devez utiliser le mot de passe de l'utilisateur principal.

   Contraintes du mot de passe :

   - Entre 8 et 100 caractères ASCII imprimables.
   - Tous les caractères ASCII imprimables peuvent être utilisés, à l'exception des suivants :
     - **/** (barre oblique)
     - **"** (guillemets doubles)
     - **@** (symbole arobase)

7. Au bas de l'écran, choisissez l'une des actions suivantes :

   - Pour créer le cluster maintenant, choisissez Create cluster (Créer un cluster).
   - Pour ne pas créer de cluster, choisissez Cancel (Annuler).
   - Pour configurer plus précisément le cluster avant de le créer, choisissez Show additional configurations (Afficher les configurations supplémentaires), puis poursuivez avec la page Création d'un cluster : configurations supplémentaires.

     Les configurations couvertes dans la section Additional Configurations (Configurations supplémentaires) sont :

     - Paramètres réseau : le groupe de sécurité `default` VPC est utilisé par défaut.
     - Options du cluster : le port par défaut est 27017 et le groupe de paramètres par défaut.
     - Chiffrement : le chiffrement est activé par défaut à l'aide de la `(default) aws/rds` clé.

     > **Important**
     >
     > Une fois qu'un cluster est chiffré, il ne peut pas être déchiffré.

- Sauvegarde — La valeur par défaut est de conserver les sauvegardes pendant 1 jour et de laisser Amazon DocumentDB choisir la fenêtre de sauvegarde.
- Exportations de journaux : par défaut, les journaux d'audit ne sont pas exportés vers CloudWatch Logs.
- Maintenance — Par défaut, Amazon DocumentDB choisit la fenêtre de maintenance.
- Protection contre la suppression : protégez votre cluster contre toute suppression accidentelle. La valeur par défaut pour les clusters créés à l'aide de la console est enabled (activé).

Si vous acceptez les paramètres par défaut maintenant, vous pouvez modifier la plupart d'entre elles ultérieurement en modifiant le cluster.

8. Activez la connexion entrante pour le groupe de sécurité de votre cluster.

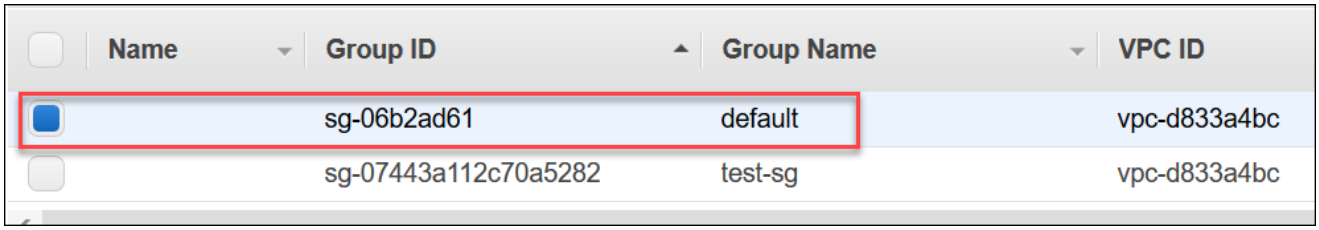
   Si vous n'avez pas modifié les paramètres par défaut de votre cluster, vous avez créé un cluster à l'aide du groupe de sécurité par défaut pour le VPC par défaut dans la région donnée. Pour vous connecter à Amazon DocumentDB, vous devez activer les connexions entrantes sur le port 27017 (ou le port de votre choix) pour le groupe de sécurité de votre cluster.

   Pour ajouter une connexion entrante au groupe de sécurité de votre cluster

   a. Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/ec2/.

   b. Dans la section Resources (Ressources) de la fenêtre principale, choisissez Security groups (Groupes de sécurité).

   c. Dans la liste des groupes de sécurité, localisez celui que vous avez utilisé lors de la création de votre cluster (il s'agit probablement du groupe de sécurité par défaut) et choisissez le champ à gauche de son nom.



d. Dans le menu Actions, choisissez Edit inbound rules (Modifier les règles entrantes), puis choisissez ou saisissez les contraintes de règle.

   i. Type —Dans la liste, choisissez le protocole à ouvrir au trafic réseau.

   ii. Protocole —Dans la liste, choisissez le type de protocole.

   iii. Plage de ports : pour une règle personnalisée, entrez un numéro de port ou une plage de ports. Veillez à ce que le numéro de port ou la plage des ports comprenne le port spécifié lors de la création de votre cluster (par défaut, 27017).

   iv. Source —Spécifie le trafic qui peut atteindre votre instance. Dans la liste, choisissez le trafic source. Si vous choisissez Custom (Personnalisé), spécifiez une adresse IP unique ou une plage d'adresses IP dans une notation CIDR (par exemple, 203.0.113.5/32).

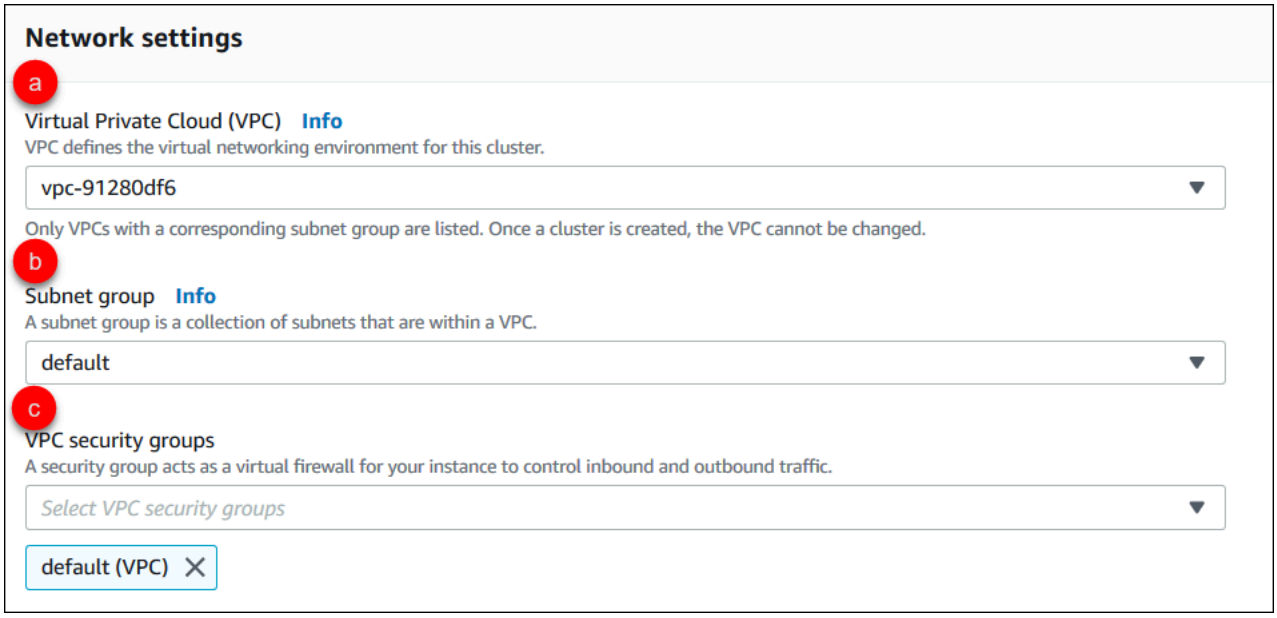
   v. Description —Entrez une description pour cette règle.

   vi. Lorsque vous avez terminé de créer la règle, choisissez Save (Enregistrer).

Création d'un cluster : configurations supplémentaires

Si vous souhaitez accepter les paramètres par défaut pour votre cluster, vous pouvez ignorer les étapes suivantes et choisir Create cluster (Créer un cluster).

1. Complétez le volet Network settings (Paramètres de réseau).



a. Virtual Private Cloud (VPC) —Dans la liste, choisissez l'Amazon VPC dans lequel vous souhaitez lancer ce cluster.

b. Groupe de sous-réseaux : dans la liste, choisissez le groupe de sous-réseaux que vous souhaitez utiliser pour ce cluster.

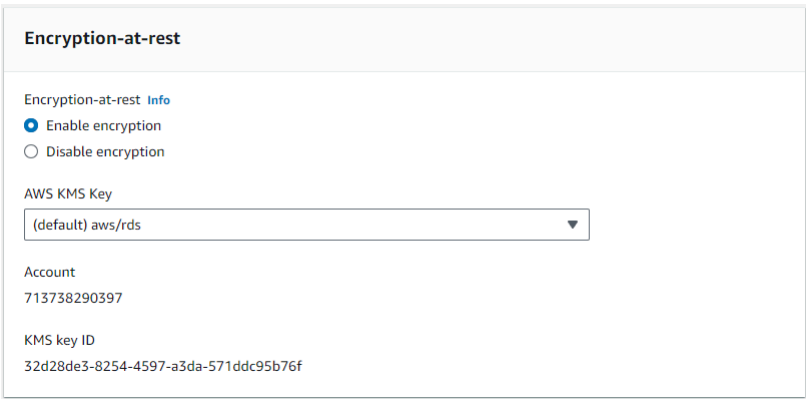
c. Groupes de sécurité VPC : dans la liste, choisissez le groupe de sécurité VPC pour ce cluster.

2. Complétez le volet Cluster options (Options du cluster).

a. Port de base de données : utilisez les flèches haut et bas pour définir le port TCP/IP que les applications utiliseront pour se connecter à votre instance.

b. Groupe de paramètres de cluster : dans la liste des groupes de paramètres, choisissez le groupe de paramètres de cluster pour ce cluster.

3. Complétez le volet Encryption (Chiffrement).



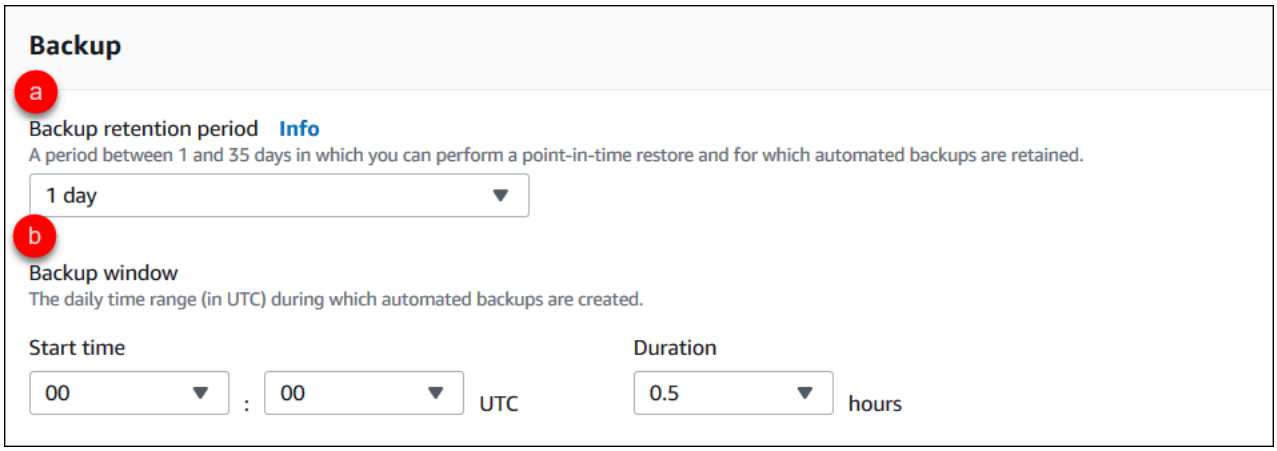
a. E ncryption-at-rest —Choisissez l'une des options suivantes :

   - Activer le chiffrement : valeur par défaut. Toutes les données au repos sont chiffrées. Si vous choisissez de chiffrer vos données, vous ne pouvez pas annuler cette action.
   - Désactiver le chiffrement : vos données ne sont pas chiffrées.

b. AWS Clé KMS : elle n'est disponible que si vous cryptez vos données. Dans la liste, choisissez la clé que vous souhaitez utiliser pour chiffrer les données de ce cluster. L'argument par défaut est `(default) aws/rds`.

   Si vous choisissez Enter a key ARN (Entrer un ARN de clé), vous devez indiquer un Amazon Resource Name (ARN) pour la clé.

4. Complétez le volet Backup (Sauvegarde).



a. Période de conservation des sauvegardes : dans la liste, choisissez le nombre de jours pendant lesquels vous pouvez conserver les sauvegardes automatiques de ce cluster avant de les supprimer.

b. Fenêtre de sauvegarde : définissez l'heure et la durée quotidiennes pendant lesquelles Amazon DocumentDB doit effectuer des sauvegardes de ce cluster.

i. Heure de début : dans la première liste, choisissez l'heure de début (UTC) pour démarrer vos sauvegardes automatiques. Dans la deuxième liste, choisissez la minute de l'heure à laquelle vous voulez que les sauvegardes automatiques commencent.

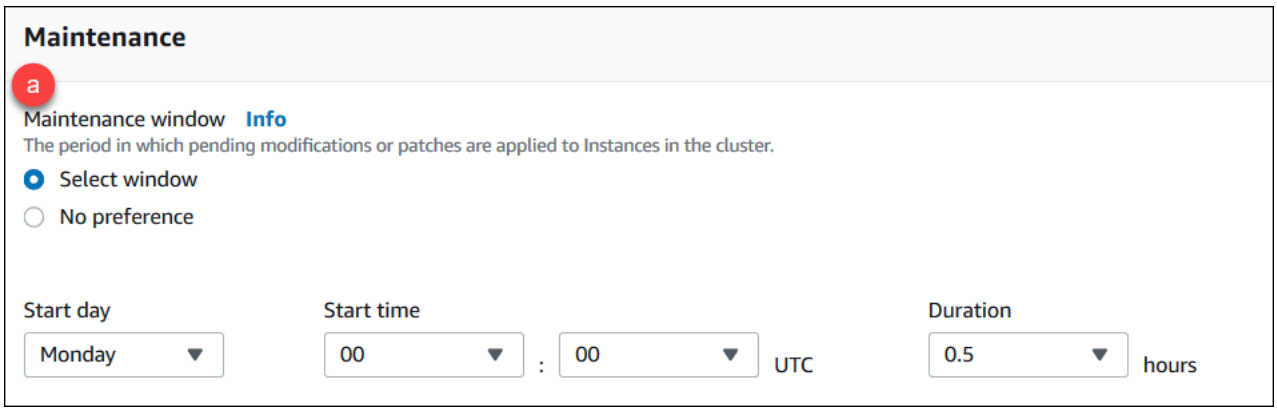
ii. Durée : dans la liste, choisissez le nombre d'heures à allouer à la création de sauvegardes automatiques.

5. Complétez le volet Exportations de journaux en sélectionnant les types de journaux que vous souhaitez exporter vers CloudWatch Logs.

- Journaux d'audit : sélectionnez cette option pour activer l'exportation des journaux d'audit vers Amazon CloudWatch Logs. Si vous sélectionnez Audit logs (Journaux d'audit), vous devez activer `audit_logs` dans le groupe de paramètres personnalisés du cluster. Pour plus d'informations, consultez Audit des événements Amazon DocumentDB.
- Journaux du profileur : sélectionnez cette option pour activer l'exportation des journaux du profileur d'opérations vers Amazon Logs. CloudWatch Si vous sélectionnez Profiler logs (Journaux du profileur), vous devez également modifier les paramètres suivants dans le groupe de paramètres personnalisés du cluster :
  - `profiler`—Réglé sur. `enabled`
  - `profiler_threshold_ms`—Définissez une valeur `[0-INT_MAX]` pour définir le seuil pour les opérations de profilage.
  - `profiler_sampling_rate`: définissez une valeur pour `[0.0-1.0]` définir le pourcentage d'opérations lentes par rapport au profil.

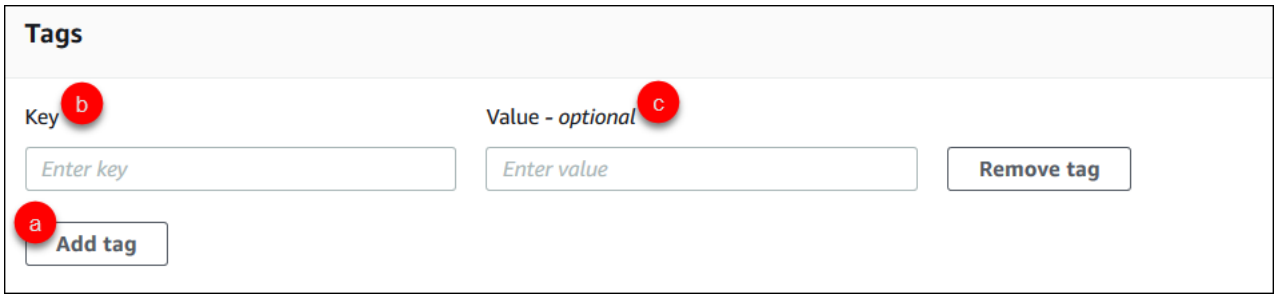
  Pour plus d'informations, consultez Profilage des opérations Amazon DocumentDB.

6. Complétez le volet Maintenance.



- Choisissez l'une des options suivantes
  - Sélectionnez une fenêtre : vous pouvez spécifier le jour de la semaine, l'heure de début UTC et la durée pendant laquelle Amazon DocumentDB doit effectuer la maintenance de votre cluster.
    a. Jour de début : dans la liste, choisissez le jour de la semaine pour démarrer la maintenance du cluster.
    b. Heure de début : dans les listes, choisissez l'heure et la minute (UTC) pour démarrer la maintenance.
    c. Durée : dans la liste, choisissez le temps à allouer à la maintenance du cluster. Si la maintenance ne peut pas être effectuée dans le laps de temps spécifié, le processus se poursuit au-delà de la durée spécifiée jusqu'à ce qu'il se termine.
  - Aucune préférence : Amazon DocumentDB choisit le jour de la semaine, l'heure de début et la durée de la maintenance.

7. Si vous souhaitez ajouter une ou plusieurs balises à ce cluster, complétez le volet Tags (Balises).



Pour chaque balise que vous souhaitez ajouter au cluster, répétez les étapes suivantes. Vous pouvez en avoir jusqu'à 10 sur un cluster.

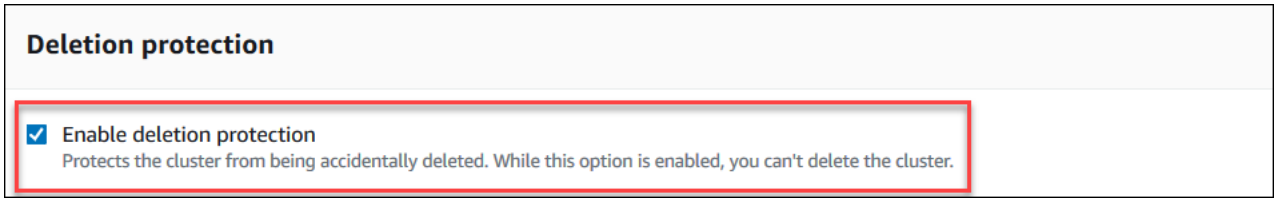
a. Sélectionnez Add Tags (Ajouter des balises).

b. Entrez la clé de la balise.

c. Vous pouvez, si vous le souhaitez, entrer la valeur de la balise.

Pour supprimer une balise, choisissez Remove tag (Supprimer une balise).

8. Deletion Protection (Protection contre la suppression) est activée par défaut lorsque vous créez un cluster à l'aide de la console. Pour désactiver la protection contre la suppression, désactivez Enable deletion protection (Activer la protection contre la suppression). Lorsque cette option est activée, la protection contre la suppression empêche la suppression d'un cluster. Pour supprimer un cluster protégé contre la suppression, vous devez tout d'abord modifier le cluster pour désactiver la protection contre la suppression.



Pour plus d'informations sur la protection de la suppression, consultez Suppression d'un cluster Amazon DocumentDB.

9. Pour créer le cluster, choisissez Create cluster (Créer un cluster). Sinon, sélectionnez Annuler.

Création d'un cluster à l'aide du AWS CLI

Les procédures suivantes décrivent comment utiliser le AWS CLI pour lancer un cluster Amazon DocumentDB et créer une réplique Amazon DocumentDB.

Paramètres

- **`--db-cluster-identifier`** : obligatoire. Une chaîne en minuscules qui identifie ce cluster.

  Contraintes d'attribution de nom relatives à un cluster :

  - La longueur est de [1 à 63] lettres, chiffres ou traits d'union.
  - Le premier caractère doit être une lettre.
  - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

  - Doit être unique pour tous les clusters (sur Amazon RDS, Amazon Neptune et Amazon DocumentDB) AWS par compte et par région.

- **`--engine`** : obligatoire. Doit indiquer **`docdb`**.
- **`--deletion-protection | --no-deletion-protection`**—Facultatif. Lorsque la protection contre la suppression est activée, elle empêche la suppression d'un cluster. Lorsque vous utilisez le AWS CLI, le paramètre par défaut est de désactiver la protection contre la suppression.

  Pour plus d'informations sur la protection de la suppression, consultez Suppression d'un cluster Amazon DocumentDB.
- **`--storage-type standard | iopt1`**—Facultatif. Par défaut: **`standard`**. Configuration de stockage du cluster. Les valeurs valides sont `standard` (Standard) ou `iopt1` (optimisées pour les E/S).
- **`--master-username`** : obligatoire. Nom de l'utilisateur authentifié.

  Contraintes d'attribution de nom d'utilisateur principal :

  - Entre 1 et 63 caractères alphanumériques.
  - Le premier caractère doit être une lettre.
  - Ne peut pas être un mot réservé du moteur de base de données.
- **`--master-user-password`** : obligatoire. Mot de passe de l'utilisateur authentifié.

  Contraintes du mot de passe principal :

  - Entre 8 et 100 caractères ASCII imprimables.
  - Tous les caractères ASCII imprimables peuvent être utilisés, à l'exception des suivants :
    - **`/`** (barre oblique)
    - **`"`** (guillemets doubles)
    - **`@`** (symbole arobase)

Pour accéder à des paramètres supplémentaires, consultez CreateDBCluster.

Pour lancer un cluster Amazon DocumentDB à l'aide du AWS CLI

Pour créer un cluster Amazon DocumentDB, appelez le. `create-db-cluster` AWS CLI La AWS CLI commande suivante crée un cluster Amazon DocumentDB nommé `sample-cluster` avec la protection contre la suppression activée. Pour plus d'informations sur la protection contre les suppressions, consultezSuppression d'un cluster Amazon DocumentDB.

Il s'`--engine-version`agit également d'un paramètre facultatif qui utilise par défaut la dernière version majeure du moteur. La version majeure actuelle du moteur est 4.0.0. Lorsque de nouvelles versions majeures du moteur sont publiées, la version par défaut du moteur est mise à jour pour `--engine-version` refléter la dernière version du moteur principal. Par conséquent, pour les charges de travail de production, et en particulier celles qui dépendent de scripts, d'automatisation ou de AWS CloudFormation modèles, nous vous recommandons de spécifier explicitement la `--engine-version` version majeure prévue.

> **Note**
>
> Si un `db-subnet-group-name` ou n'`vpc-security-group-id`est pas spécifié, Amazon DocumentDB utilisera le groupe de sous-réseaux et le groupe de sécurité Amazon VPC par défaut pour la région donnée.

Pour Linux, macOS ou Unix :

```
aws docdb create-db-cluster \
      --db-cluster-identifier sample-cluster \
      --engine docdb \
      --engine-version 4.0.0 \
      --deletion-protection \
      --master-username masteruser \
      --master-user-password password
```

Pour Windows :

```
aws docdb create-db-cluster ^
      --db-cluster-identifier sample-cluster ^
      --engine docdb ^
      --engine-version 4.0.0 ^
      --deletion-protection ^
      --master-username masteruser ^
      --master-user-password password
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "DBCluster": {
        "StorageEncrypted": false,
        "DBClusterMembers": [],
        "Engine": "docdb",
        "DeletionProtection" : "enabled",
        "ClusterCreateTime": "2018-11-26T17:15:19.885Z",
        "DBSubnetGroup": "default",
        "EngineVersion": "4.0.0",
        "MasterUsername": "masteruser",
        "BackupRetentionPeriod": 1,
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
        "DBClusterIdentifier": "sample-cluster",
        "MultiAZ": false,
        "DBClusterParameterGroup": "default.docdb4.0",
        "PreferredBackupWindow": "09:12-09:42",
        "DbClusterResourceId": "cluster-KQSGI4MHU4NTDDRVNLNTU7XVAY",
        "PreferredMaintenanceWindow": "tue:04:17-tue:04:47",
        "Port": 27017,
        "Status": "creating",
        "ReaderEndpoint": "sample-cluster.cluster-ro-sfcrlcjcoroz.us-
east-1.docdb.amazonaws.com",
        "AssociatedRoles": [],
        "HostedZoneId": "ZNKXTT8WH85VW",
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1e"
        ],
        "Endpoint": "sample-cluster.cluster-sfcrlcjcoroz.us-east-1.docdb.amazonaws.com"
    }
}
```

La création du cluster prend quelques minutes. Vous pouvez utiliser le AWS Management Console ou AWS CLI pour surveiller l'état de votre cluster. Pour plus d'informations, consultez Surveillance de l'état d'un cluster Amazon DocumentDB.

> **Important**
>
> Lorsque vous utilisez le AWS CLI pour créer un cluster Amazon DocumentDB, aucune instance n'est créée. Par conséquent, vous devez créer explicitement une instance principale et tous les réplicas des instances dont vous avez besoin. Vous pouvez utiliser la console ou AWS CLI créer les instances. Pour plus d'informations, consultez Ajouter une instance Amazon DocumentDB à un cluster.

Pour plus d'informations, consultez le `CreateDBCluster`manuel Amazon DocumentDB API Reference.

Décrire les clusters Amazon DocumentDB

Vous pouvez utiliser la console de gestion Amazon DocumentDB ou le AWS CLI pour consulter des informations telles que les points de terminaison de connexion, les groupes de sécurité, les VPC et les groupes de paramètres relatifs à vos clusters Amazon DocumentDB.

Pour plus d'informations, consultez les ressources suivantes :

- Surveillance de l'état d'un cluster Amazon DocumentDB
- Recherche des points de terminaison d'un cluster

Using the AWS Management Console

Utilisez la procédure suivante pour afficher les détails d'un cluster Amazon DocumentDB spécifié à l'aide de la console.

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans la liste des clusters, choisissez le nom du cluster dont vous voulez afficher les détails. Les informations sur le cluster sont organisées en six groupes :

   - Résumé : informations générales sur le cluster, notamment la version du moteur, l'état du cluster, la maintenance en attente et le statut de son groupe de paramètres.
   - Connectivité et sécurité : la section Connect répertorie les points de terminaison de connexion permettant de se connecter à ce cluster à l'aide du shell mongo ou d'une application. La section Security Groups (Groupes de sécurité) répertorie les groupes de sécurité associés à ce cluster ainsi que leur ID de VPC et leurs descriptions.
   - Configuration : la section Détails du cluster répertorie les informations relatives au cluster, notamment le nom de ressource Amazon (ARN), le point de terminaison et le groupe de paramètres du cluster. Elle répertorie également les informations de sauvegarde du cluster, les détails de maintenance ainsi que les paramètres de sécurité et de réseau. La section Cluster instances (Instances de cluster) répertorie toutes les instances appartenant à votre cluster avec l'état du rôle et du groupe de paramètres de cluster de chaque instance.
   - Surveillance — Les métriques Amazon CloudWatch Logs pour ce cluster. Pour plus d'informations, consultez Surveillance d'Amazon DocumentDB avec CloudWatch.
   - Événements et tags — La section Événements récents répertorie les événements récents pour ce cluster. Amazon DocumentDB conserve un enregistrement des événements liés à vos clusters, instances, instantanés, groupes de sécurité et groupes de paramètres de cluster. Ces informations comprennent la date, l'heure et le message associés à chaque événement. La section Tags (Balises) répertorie les balises attachées à ce cluster.

Using the AWS CLI

Pour afficher les détails de vos clusters Amazon DocumentDB à l'aide de AWS CLI, utilisez la `describe-db-clusters` commande comme indiqué dans les exemples ci-dessous. Pour plus d'informations, consultez le document `DescribeDBClusters`de référence de l'API de gestion des ressources Amazon DocumentDB.

> **Note**
>
> Pour certaines fonctionnalités de gestion telles que la gestion du cycle de vie des clusters et des instances, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon RDS. Le paramètre de `filterName=engine,Values=docdb` filtre renvoie uniquement les clusters Amazon DocumentDB.

Example

Exemple 1 : répertorier tous les clusters Amazon DocumentDB

Le AWS CLI code suivant répertorie les détails de tous les clusters Amazon DocumentDB d'une région.

```
aws docdb describe-db-clusters --filter Name=engine,Values=docdb
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBClusters": [
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-1",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        },
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-2",
```

```
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        },
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-3",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        }
    ]
}
```

Example

Exemple 2 : répertorier tous les détails d'un cluster Amazon DocumentDB spécifié

Le AWS CLI code suivant répertorie les détails du cluster`sample-cluster`.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-clusters \
   --filter Name=engine,Values=docdb \
   --db-cluster-identifier sample-cluster
```

Pour Windows :

```
aws docdb describe-db-clusters ^
   --filter Name=engine,Values=docdb ^
   --db-cluster-identifier sample-cluster
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBClusters": [
```

```
        {
            "AllocatedStorage": 1,
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1a",
                "us-east-1d"
            ],
            "BackupRetentionPeriod": 2,
            "DBClusterIdentifier": "sample-cluster",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            "EarliestRestorableTime": "2023-11-07T22:34:08.148000+00:00",
            "Endpoint": "sample-cluster.node.us-east-1.amazon.com",
            "ReaderEndpoint": "sample-cluster.node.us-east-1.amazon.com",
            "MultiAZ": false,
            "Engine": "docdb",
            "EngineVersion": "5.0.0",
            "LatestRestorableTime": "2023-11-10T07:21:16.772000+00:00",
            "Port": 27017,
            "MasterUsername": "chimeraAdmin",
            "PreferredBackupWindow": "22:22-22:52",
            "PreferredMaintenanceWindow": "sun:03:01-sun:03:31",
            "ReadReplicaIdentifiers": [],
            "DBClusterMembers": [
                {
                    "DBInstanceIdentifier": "sample-instance-1",
                    "IsClusterWriter": true,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },
                {
                    "DBInstanceIdentifier": "sample-instance-2",
                    "IsClusterWriter": true,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },

            ],
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-9084c2ec",
                    "Status": "active"
                }
```

```
            ],
            "HostedZoneId": "Z06853723JYKYBXTJ49RB",
            "StorageEncrypted": false,
            "DbClusterResourceId": "cluster-T4LGLANHVAPGQYYULWUDKLVQL4",
            "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-
cluster",
            "AssociatedRoles": [],
            "IAMDatabaseAuthenticationEnabled": false,
            "ClusterCreateTime": "2023-11-06T18:05:41.568000+00:00",
            "EngineMode": "provisioned",
            "DeletionProtection": false,
            "HttpEndpointEnabled": false,
            "CopyTagsToSnapshot": false,
            "CrossAccountClone": false,
            "DomainMemberships": [],
            "TagList": [],
            "StorageType": "iopt1",
            "AutoMinorVersionUpgrade": false,
            "NetworkType": "IPV4",
            "IOOptimizedNextAllowedModificationTime":
 "2023-12-07T18:05:41.580000+00:00"
        }
    ]
}
```

Example

Exemple 3 : Répertorier les informations spécifiques d'un cluster Amazon DocumentDB

Pour répertorier un sous-ensemble des détails des clusters à l'aide du AWS CLI, ajoutez un qui spécifie les membres du cluster `--query` que l'`describe-db-clusters`opération doit répertorier. Le paramètre `--db-cluster-identifier` est l'identifiant du cluster particulier dont vous souhaitez afficher les détails. Pour plus d'informations sur les requêtes, voir Comment filtrer la sortie avec l'`--query`option dans le guide de AWS Command Line Interface l'utilisateur.

L'exemple suivant répertorie les instances d'un cluster Amazon DocumentDB.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-clusters \
    --filter Name=engine,Values=docdb \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].[DBClusterMembers]'
```

Pour Windows :

```
aws docdb describe-db-clusters ^
    --filter Name=engine,Values=docdb ^
    --db-cluster-identifier sample-cluster ^
    --query 'DBClusters[*].[DBClusterMembers]'
```

Le résultat de cette opération ressemble à ceci.

```
[
    [
        [
            {
                "DBInstanceIdentifier": "sample-instance-1",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            },
            {
                "DBInstanceIdentifier": "sample-instance-2",
                "IsClusterWriter": false,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ]
    ]
]
```

Modification d'un cluster Amazon DocumentDB

Pour modifier un cluster, le cluster doit être à l'état disponible . Vous ne pouvez pas modifier un cluster qui est arrêté. Si le cluster est arrêté, commencez par démarrer le cluster, attendez que le cluster devienne disponible, puis apportez les modifications souhaitées. Pour plus d’informations, consultez Arrêt et démarrage d'un cluster Amazon DocumentDB.

Using the AWS Management Console

Utilisez la procédure suivante pour modifier un cluster Amazon DocumentDB spécifique à l'aide de la console.

Pour modifier un cluster Amazon DocumentDB

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Spécifiez le cluster que vous souhaitez modifier en cliquant sur le bouton situé à gauche du nom du cluster.
4. Choisissez Actions, puis Modify (Modifier).
5. Dans le volet Modify cluster: <cluster-name> (Modifier le cluster : <nom-cluster>), apportez les modifications souhaitées. Vous pouvez effectuer des modifications dans les domaines suivants :

   - Spécifications du cluster : nom, groupes de sécurité et mot de passe du cluster.
   - Configuration du stockage en cluster : mode de stockage des données du cluster. Choisissez entre une configuration standard et une configuration optimisée pour les E/S.
   - Options du cluster : port et groupe de paramètres du cluster.
   - Sauvegarde : période de conservation des sauvegardes et fenêtre de sauvegarde du cluster.
   - Exportations de journaux : activez ou désactivez l'exportation des journaux d'audit ou de profilage.
   - Maintenance —Définissez la fenêtre de maintenance du cluster.
   - Protection contre la suppression : activez ou désactivez la protection contre la suppression sur le cluster. Par défaut, la protection contre la suppression est activée.

6. Lorsque vous avez terminé, choisissez Continue (Continuer) pour afficher un récapitulatif de vos modifications.

7. Si vous êtes satisfait de vos modifications, vous pouvez choisir Modify cluster (Modifier le cluster) pour modifier votre cluster. Vous pouvez également choisir Back (Précédent) ou Cancel (Annuler) pour modifier ou annuler vos modifications, respectivement.

L'application de vos modifications prend quelques minutes. Vous pouvez uniquement utiliser le cluster lorsqu'il présente le statut disponible. Vous pouvez surveiller l'état du cluster en utilisant la console ou la AWS CLI. Pour plus d'informations, consultez Surveillance de l'état d'un cluster Amazon DocumentDB.

Using the AWS CLI

Utilisez l'opération `modify-db-cluster` pour modifier le cluster spécifié à l'aide de l' AWS CLI. Pour plus d'informations, consultez le `ModifyDBCluster`manuel Amazon DocumentDB API Reference.

Paramètres

- **`--db-cluster-identifier`** : obligatoire. Identifiant du cluster Amazon DocumentDB que vous allez modifier.
- **`--backup-retention-period`**—Facultatif. Nombre de jours de conservation des sauvegardes automatiques. Les valeurs valides sont comprises entre 1 et 35.
- **`--storage-type`**—Facultatif. Configuration de stockage du cluster. Les valeurs valides sont `standard` (Standard) ou `iopt1` (optimisées pour les E/S).
- **`--db-cluster-parameter-group-name`**—Facultatif. Le nom du groupe de paramètres de cluster à utiliser pour le cluster.
- **`--master-user-password`**—Facultatif. Le nouveau mot de passe de l'utilisateur principal de la base de données.

  Contraintes du mot de passe :

  - La longueur est de [8 à 100] caractères ASCII imprimables.
  - Tous les caractères ASCII imprimables peuvent être utilisés, à l'exception des suivants :
    - **`/`** (barre oblique)
    - **`"`** (guillemets doubles)
    - **`@`** (symbole arobase)

- **`--new-db-cluster-identifier`**—Facultatif. Le nouvel identificateur de cluster pour le cluster lors du changement de nom d'un cluster. Cette valeur est stockée sous la forme d'une chaîne en minuscules.

  Contraintes d'affectation de noms :

  - La longueur est de [1 à 63] lettres, chiffres ou traits d'union.
  - Le premier caractère doit être une lettre.
  - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
  - Doit être unique pour tous les clusters d'Amazon RDS, Amazon Neptune et Amazon DocumentDB par région Compte AWS.
- **`--preferred-backup-window`**—Facultatif. L'intervalle de temps quotidien (en UTC) au cours duquel les sauvegardes automatiques sont créées.
  - Format : `hh24:mm-hh24:mm`
- **`--preferred-maintenance-window`**—Facultatif. Intervalle de temps hebdomadaire (en UTC) pendant lequel peut se produire la maintenance du système.
  - Format : `ddd:hh24:mm-ddd:hh24:mm`
  - Jours valables : `Sun`, `Mon`, `Tue`, `Wed`, `Thu`, `Fri`, et `Sat`.
- **`--deletion-protection`**ou **`--no-deletion-protection`** —Facultatif. Indique si la protection contre la suppression doit être activée sur ce cluster. La protection contre la suppression empêche toute suppression accidentelle d'un cluster tant que cette fonction n'est pas désactivée. Pour plus d'informations, consultez Suppression d'un cluster Amazon DocumentDB.
- **`--apply-immediately`**ou **`--no-apply-immediately`** —Utilisez `--apply-immediately` pour effectuer la modification immédiatement. Utilisez `--no-apply-immediately` pour effectuer la modification au cours de la fenêtre de maintenance suivante de votre cluster.

Example

Le code suivant modifie la période de conservation des sauvegardes pour le cluster `sample-cluster`.

Pour Linux, macOS ou Unix :

```
aws docdb modify-db-cluster \
       --db-cluster-identifier sample-cluster \
```

```
       --apply-immediately \
       --backup-retention-period 7
```

Pour Windows :

```
aws docdb modify-db-cluster ^
       --db-cluster-identifier sample-cluster ^
       --apply-immediately ^
       --backup-retention-period 7
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBCluster": {
        "BackupRetentionPeriod": 7,
        "DbClusterResourceId": "cluster-VDP53QEWST7YHM36TTXOPJT5YE",
        "Status": "available",
        "DBClusterMembers": [
            {
                "PromotionTier": 1,
                "DBClusterParameterGroupStatus": "in-sync",
                "DBInstanceIdentifier": "sample-cluster-instance",
                "IsClusterWriter": true
            }
        ],
        "ReadReplicaIdentifiers": [],
        "AvailabilityZones": [
            "us-east-1b",
            "us-east-1c",
            "us-east-1a"
        ],
        "ReaderEndpoint": "sample-cluster.cluster-ro-ctevjxdlur57.us-
east-1.rds.amazonaws.com",
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
        "PreferredMaintenanceWindow": "sat:09:51-sat:10:21",
        "EarliestRestorableTime": "2018-06-17T00:06:19.374Z",
        "StorageEncrypted": false,
        "MultiAZ": false,
        "AssociatedRoles": [],
        "MasterUsername": "<your-master-user-name>",
        "DBClusterIdentifier": "sample-cluster",
        "VpcSecurityGroups": [
            {
```

```
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "HostedZoneId": "Z2SUY0A1719RZT",
        "LatestRestorableTime": "2018-06-18T21:17:05.737Z",
        "AllocatedStorage": 1,
        "Port": 27017,
        "Engine": "docdb",
        "DBClusterParameterGroup": "default.docdb3.4",
        "Endpoint": "sample-cluster.cluster-ctevjxdlur57.us-
east-1.rds.amazonaws.com",
        "DBSubnetGroup": "default",
        "PreferredBackupWindow": "00:00-00:30",
        "EngineVersion": "3.4",
        "ClusterCreateTime": "2018-06-06T19:25:47.991Z",
        "IAMDatabaseAuthenticationEnabled": false
    }
}
```

L'application de vos modifications prend quelques minutes. Vous pouvez uniquement utiliser le cluster lorsqu'il présente le statut disponible. Vous pouvez surveiller l'état du cluster en utilisant la console ou la AWS CLI. Pour plus d'informations, consultez Surveillance de l'état d'un cluster Amazon DocumentDB.

Détermination de la maintenance en attente

Vous pouvez déterminer si vous disposez de la dernière version du moteur Amazon DocumentDB en déterminant si vous avez une maintenance de cluster en attente.

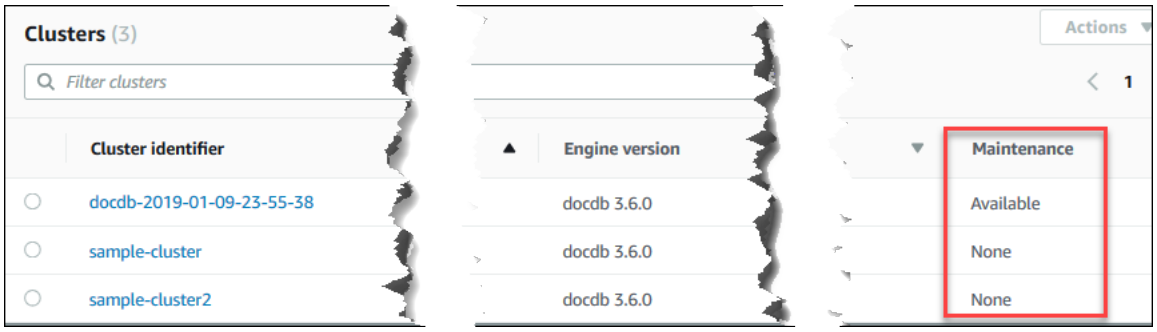
Using the AWS Management Console

Vous pouvez utiliser le AWS Management Console pour déterminer si un cluster a une maintenance en attente.

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Recherchez la colonne Maintenance afin de déterminer si un cluster est en attente de maintenance.



   None (Aucune) indique que le cluster exécute la dernière version du moteur. Available (Disponible) indique que le cluster est en attente de la maintenance, ce qui implique qu'une mise à niveau du moteur est nécessaire.

4. Si votre cluster est en attente de maintenance, poursuivez avec les étapes de la section Exécution d'une mise à jour du correctif de la version du moteur d'un cluster.

Using the AWS CLI

Vous pouvez utiliser le AWS CLI pour déterminer si un cluster possède la dernière version du moteur en utilisant l'`describe-pending-maintenance-action`opération avec les paramètres suivants.

Paramètres

- **`--resource-identifier`**—Facultatif. ARN d'accès à la ressource (cluster). Si ce paramètre n'est pas spécifié, les actions de maintenance en attente pour tous les clusters sont répertoriées.
- **`--region`**—Facultatif. La région AWS dans laquelle vous voulez exécuter cette opération, par exemple `us-east-1`.

Example

Pour Linux, macOS ou Unix :

```
aws docdb describe-pending-maintenance-actions \
   --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster \
   --region us-east-1
```

Pour Windows :

```
aws docdb describe-pending-maintenance-actions ^
   --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster ^
   --region us-east-1
```

Le résultat de cette opération ressemble à ceci.

```
{
    "PendingMaintenanceActions": [
        {
            "ResourceIdentifier": "arn:aws:rds:us-
east-1:123456789012:cluster:sample-cluster",
            "PendingMaintenanceActionDetails": [
                {
                    "Description": "New feature",
                    "Action": "db-upgrade",
                    "ForcedApplyDate": "2019-02-25T21:46:00Z",
                    "AutoAppliedAfterDate": "2019-02-25T07:41:00Z",
                    "CurrentApplyDate": "2019-02-25T07:41:00Z"
                }
            ]
        }
    ]
}
```

Si votre cluster est en attente de maintenance, poursuivez avec les étapes de la section Exécution d'une mise à jour du correctif de la version du moteur d'un cluster.

Exécution d'une mise à jour du correctif de la version du moteur d'un cluster

Dans cette section, nous expliquerons comment déployer une mise à jour de correctif à l'aide du AWS Management Console ou du AWS CLI. Une mise à jour de correctif est une mise à jour de la même version du moteur (par exemple, la mise à jour d'une version du moteur 3.6 vers une version du moteur 3.6 plus récente). Vous pouvez le mettre à jour immédiatement ou lors de la prochaine fenêtre de maintenance de votre cluster. Pour déterminer si votre moteur a besoin d'une mise à jour, consultezDétermination de la maintenance en attente. Notez que lorsque vous appliquez la mise à jour, votre cluster subira des temps d'arrêt.

> **Note**
>
> Si vous essayez de passer d'une version majeure du moteur à une autre, par exemple de la version 3.6 à la version 5.0, reportez-vous à la section Mise à niveau sur place de la version majeure d'Amazon DocumentDB ouMise à niveau de votre cluster Amazon DocumentDB à l'aide de AWS Database Migration Service. Une mise à niveau de version majeure sur place ne prend en charge que docdb 5.0 en tant que version du moteur cible.

Deux exigences de configuration sont requises pour obtenir les dernières mises à jour des correctifs pour la version du moteur d'un cluster :

- Le statut du cluster doit être disponible.
- Le cluster doit exécuter une version antérieure du moteur.

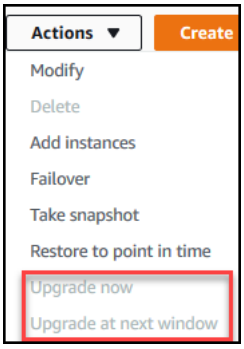
Using the AWS Management Console

La procédure suivante applique les mises à jour des correctifs à la version du moteur de votre cluster à l'aide de la console. Vous avez la possibilité de procéder à la mise à jour immédiatement ou lors de la prochaine fenêtre de maintenance de votre cluster.

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters. Dans la liste des clusters, choisissez le bouton sur la gauche du cluster que vous voulez mettre à niveau. Le statut du cluster doit être disponible.

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans le menu Actions, choisissez l'une des options suivantes. Ces options de menu sont sélectionnables uniquement si le cluster que vous avez choisi n'exécute pas la version la plus récente du moteur.



   - Effectuez la mise à niveau maintenant —Lance immédiatement le processus de mise à niveau. Votre cluster est hors ligne pendant la mise à niveau vers la version la plus récente du moteur.
   - Mise à niveau à la fenêtre suivante : lance le processus de mise à niveau lors de la fenêtre de maintenance suivante du cluster. Votre cluster est hors ligne pendant la mise à niveau vers la version la plus récente du moteur.

4. Lorsque la fenêtre de confirmation s'ouvre, choisissez l'une des options suivantes :
   - Mise à niveau : pour mettre à niveau votre cluster vers la dernière version du moteur conformément au calendrier choisi à l'étape précédente.
   - Annuler : pour annuler la mise à niveau du moteur du cluster et continuer avec la version actuelle du moteur du cluster.

Using the AWS CLI

Vous pouvez appliquer des mises à jour de correctifs à votre cluster à l'aide de l'`apply-pending-maintenance-action`opération AWS CLI et avec les paramètres suivants.

Paramètres

- **`--resource-identifier`** : obligatoire. L'ARN du cluster Amazon DocumentDB que vous allez mettre à niveau.
- **`--apply-action`** : obligatoire. Les valeurs suivantes sont autorisées. Pour mettre à niveau la version du moteur de votre cluster, utilisez `db-upgrade`.
  - **`db-upgrade`**
  - **`system-update`**
- **`--opt-in-type`** : obligatoire. Les valeurs suivantes sont autorisées.
  - `immediate`—Appliquez immédiatement l'action de maintenance.
  - `next-maintenance`—Appliquez l'action de maintenance lors de la fenêtre de maintenance suivante.
  - `undo-opt-in`—Annulez toutes les demandes d'`next-maintenance`opt-in existantes.

Example

L'exemple de correctif suivant met à jour la version du moteur de `sample-cluster` vers la version 4.0.0.

Pour Linux, macOS ou Unix :

```
aws docdb apply-pending-maintenance-action \
   --resource-identifier arn:aws:rds:us-east-1:123456789012\:cluster:sample-cluster
 \
   --apply-action db-upgrade \
   --opt-in-type immediate
```

Pour Windows :

```
aws docdb apply-pending-maintenance-action ^
   --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster ^
   --apply-action db-upgrade ^
   --opt-in-type immediate
```

Le résultat de cette opération ressemble à ce qui suit.

```
{
```

```
    "ResourcePendingMaintenanceActions": {
        "ResourceIdentifier": "arn:aws:rds:us-
east-1:444455556666:cluster:docdb-2019-01-09-23-55-38",
        "PendingMaintenanceActionDetails": [
            {
                "CurrentApplyDate": "2019-02-20T20:57:06.904Z",
                "Description": "Bug fixes",
                "ForcedApplyDate": "2019-02-25T21:46:00Z",
                "OptInStatus": "immediate",
                "Action": "db-upgrade",
                "AutoAppliedAfterDate": "2019-02-25T07:41:00Z"
            }
        ]
    }
}
```

Arrêt et démarrage d'un cluster Amazon DocumentDB

L'arrêt et le démarrage des clusters Amazon DocumentDB peuvent vous aider à gérer les coûts des environnements de développement et de test. Au lieu de créer et de supprimer des clusters et des instances à chaque fois que vous utilisez Amazon DocumentDB, vous pouvez arrêter temporairement toutes les instances de votre cluster lorsqu'elles ne sont pas nécessaires. Vous pouvez ensuite les redémarrer lorsque vous reprenez vos tests.

Rubriques

- Présentation de l'arrêt et du démarrage d'un cluster
- Opérations que vous pouvez effectuer sur un cluster arrêté

Présentation de l'arrêt et du démarrage d'un cluster

Pendant les périodes où vous n'avez pas besoin d'un cluster Amazon DocumentDB, vous pouvez arrêter toutes les instances de ce cluster en même temps. Vous pouvez ensuite à tout moment redémarrer le cluster dès que vous avez besoin de l'utiliser. Le démarrage et l'arrêt simplifie les processus de configuration et de destruction des clusters utilisés à des fins de développement, de test ou d'activités similaires qui ne nécessitent pas une disponibilité continue. Vous pouvez arrêter et démarrer un cluster en utilisant AWS Management Console ou en AWS CLI une seule action, quel que soit le nombre d'instances présentes dans le cluster.

Pendant que votre cluster est arrêté, le volume de stockage du cluster reste inchangé. Vous êtes facturé uniquement pour le stockage, les instantanés manuels et le stockage des sauvegardes automatiques pendant la fenêtre de conservation spécifiée. Aucune heure d'instance ne vous est facturée. Amazon DocumentDB démarre automatiquement votre cluster au bout de sept jours afin qu'il ne prenne aucun retard par rapport aux mises à jour de maintenance requises. Lorsque votre cluster démarre après sept jours, vous êtes à nouveau facturé pour les instances dans le cluster. Pendant l'arrêt de votre cluster, vous ne pouvez pas interroger votre volume de stockage car l'interrogation nécessite que les instances soient à l'état disponible.

Lorsqu'un cluster Amazon DocumentDB est arrêté, ni le cluster ni ses instances ne peuvent être modifiés de quelque manière que ce soit. Cela inclut l'ajout ou la suppression d'instances, ou la suppression du cluster.

Using the AWS Management Console

La procédure suivante vous montre comment arrêter un cluster avec une ou plusieurs instances à l'état disponible, ou démarrer un cluster arrêté.

Pour arrêter ou démarrer un cluster Amazon DocumentDB

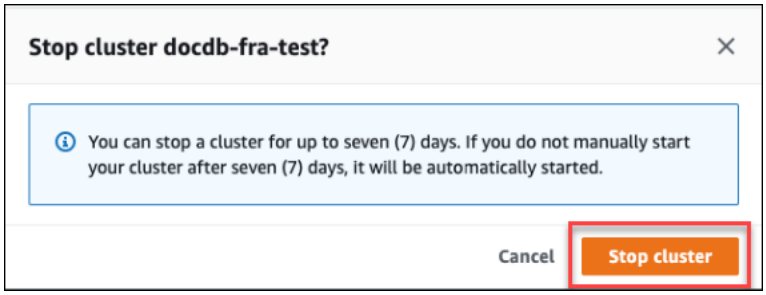
1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

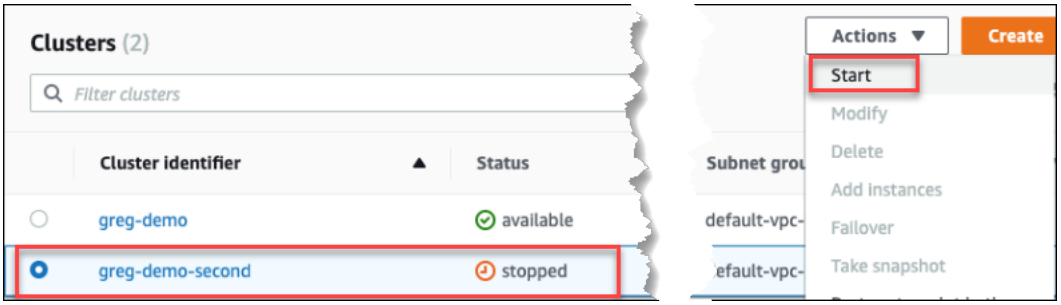
3. Dans la liste des clusters, choisissez le bouton sur la gauche du nom du cluster que vous voulez arrêter ou démarrer.
4. Choisissez Actions, puis l'action que vous souhaitez exécuter sur le cluster.
   - Si vous souhaitez arrêter le cluster et que celui-ci est disponible :
     a. Choisissez Arrêter.

Pour éviter d'activer le mécanisme de basculement, l'opération stoppe d'abord les instances de réplica, puis l'instance principale.

b. Dans la boîte de dialogue de confirmation, confirmez que vous souhaitez arrêter le cluster en choisissant Stop cluster (Arrêter le cluster). Sinon, choisissez Cancel (Annuler) pour conserver le cluster en cours d'exécution.



- Si vous souhaitez démarrer un cluster et que celui-ci est arrêté, choisissez Start (Démarrer).



5. Surveillez le statut du cluster et de ses instances. Si vous avez démarré le cluster, vous pouvez utiliser celui-ci et ses instances lorsqu'ils sont disponibles. Pour plus d'informations, consultez Déterminer le statut d'un cluster.

Using the AWS CLI

Les exemples de code suivants montrent comment arrêter un cluster avec une ou plusieurs instances à l'état disponible, ou démarrer un cluster arrêté.

Pour arrêter un cluster avec une ou plusieurs instances disponibles à l'aide de AWS CLI, utilisez l'`stop-db-cluster`opération. Pour démarrer un cluster arrêté, utilisez l'opération `start-db-cluster`. Les deux opérations utilisent le paramètre `--db-cluster-identifier`.

Paramètre :

- **`--db-cluster-identifier`** : obligatoire. Le nom du cluster à arrêter ou démarrer.

Example — Pour arrêter un cluster à l'aide du AWS CLI

Le code suivant arrête le cluster `sample-cluster`. Le cluster doit avoir une ou plusieurs instances à l'état disponible.

Pour Linux, macOS ou Unix :

```
aws docdb stop-db-cluster \
   --db-cluster-identifier sample-cluster
```

Pour Windows :

```
aws docdb stop-db-cluster ^
   --db-cluster-identifier sample-cluster
```

Example — Pour démarrer un cluster à l'aide du AWS CLI

Le code suivant démarre le cluster `sample-cluster`. Le cluster doit être actuellement à l’arrêt.

Pour Linux, macOS ou Unix :

```
aws docdb start-db-cluster \
   --db-cluster-identifier sample-cluster
```

Pour Windows :

```
aws docdb start-db-cluster ^
   --db-cluster-identifier sample-cluster
```

Opérations que vous pouvez effectuer sur un cluster arrêté

Lorsqu'un cluster Amazon DocumentDB est arrêté, vous pouvez effectuer une point-in-time restauration à tout moment pendant la période de conservation automatique des sauvegardes que vous avez spécifiée. Pour plus de détails sur la réalisation d'une point-in-time restauration, consultezRestaurez à un instant dans le passé.

Vous ne pouvez pas modifier la configuration d'un cluster Amazon DocumentDB, ou de l'une de ses instances, lorsque le cluster est arrêté. De même, vous ne pouvez pas ajouter ou supprimer des instances au niveau du cluster, ni supprimer le cluster si une ou plusieurs instances lui sont toujours associées. Vous devez démarrer le cluster avant d'effectuer des opérations d'administration de ce type.

Amazon DocumentDB applique toute maintenance planifiée à votre cluster arrêté uniquement après son redémarrage. Au bout de sept jours, Amazon DocumentDB démarre automatiquement un cluster arrêté afin qu'il ne prenne pas trop de retard dans son état de maintenance. Lorsque le cluster redémarre, vous êtes à nouveau facturé pour les instances dans le cluster.

Lorsqu'un cluster est arrêté, Amazon DocumentDB n'effectue aucune sauvegarde automatique et ne prolonge pas la période de conservation des sauvegardes.

Suppression d'un cluster Amazon DocumentDB

Vous pouvez supprimer un cluster Amazon DocumentDB à l'aide du AWS Management Console ou du. AWS CLI Pour supprimer un cluster, celui-ci doit être à l'état disponible et ne doit pas avoir d'instances associées. Si le cluster est arrêté, commencez par démarrer le cluster, attendez que le cluster devienne disponible, puis supprimez le cluster. Pour plus d’informations, consultez Arrêt et démarrage d'un cluster Amazon DocumentDB.

Deletion protection (Protection contre la suppression)

Pour protéger votre cluster de toute suppression accidentelle, vous pouvez activer la protection contre la suppression. La protection contre la suppression est activée par défaut lorsque vous créez un cluster à l'aide de la console. Elle est toutefois désactivée par défaut si vous créez un cluster à l'aide de l' AWS CLI.

Amazon DocumentDB applique la protection contre la suppression à un cluster, que vous exécutiez l'opération de suppression à l'aide de la console ou du. AWS CLI Si la protection contre la suppression est activée, vous ne pouvez pas supprimer de cluster. Pour supprimer un cluster pour lequel la protection contre la suppression est activée, vous devez commencer par modifier le cluster et désactiver la protection contre la suppression.

Lorsque vous utilisez la console avec la protection contre la suppression activée sur un cluster, vous ne pouvez pas supprimer la dernière instance du cluster car cela supprime également le cluster. Vous pouvez supprimer la dernière instance d'un cluster protégé contre la suppression à l'aide de l' AWS CLI. Toutefois, le cluster lui-même existe toujours, et vos données sont conservées. Vous pouvez accéder aux données en créant de nouvelles instances pour le cluster. Pour plus d'informations sur l'activation et la désactivation de la protection contre la suppression, consultez :

- Création d'un cluster Amazon DocumentDB
- Modification d'un cluster Amazon DocumentDB

Using the AWS Management Console

Pour supprimer un cluster à l'aide du AWS Management Console, la protection contre la suppression doit être désactivée.

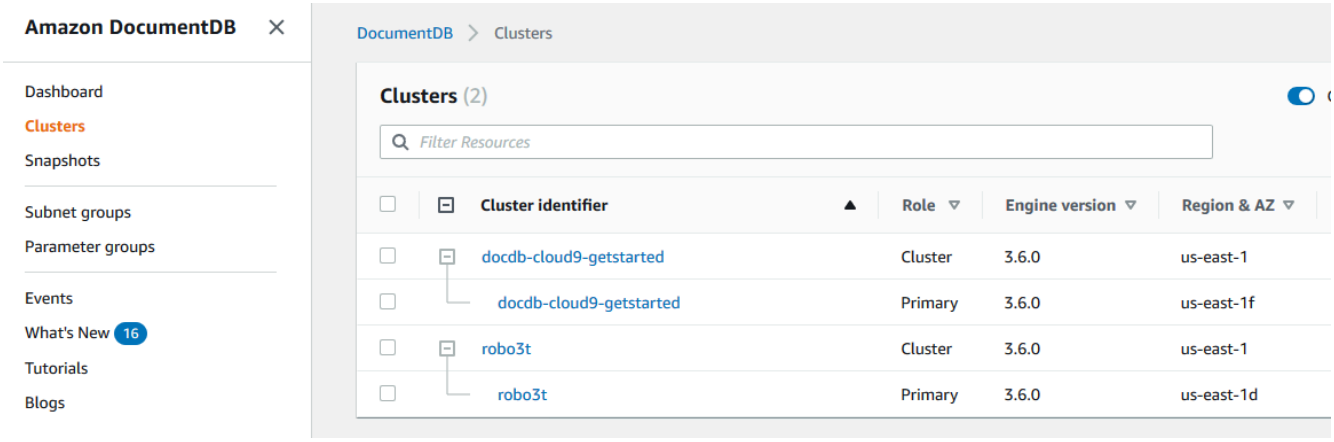
Pour déterminer si la protection contre la suppression est activée pour un cluster :

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu

(≡)
dans le coin supérieur gauche de la page.

3. Notez que dans la zone de navigation Clusters, la colonne Cluster Identifier indique à la fois les clusters et les instances. Les instances sont répertoriées sous les clusters, comme dans la capture d'écran ci-dessous.



4. Choisissez le nom du cluster et sélectionnez l'onglet Configuration . Dans la section Cluster details (Détails du cluster), recherchez Deletion protection (Protection contre la suppression). Si la protection contre la suppression est activée, modifiez le cluster pour la désactiver. Pour plus d'informations sur la modification d'un cluster, consultez Modification d'un cluster Amazon DocumentDB.

   Lorsque Deletion protection (Protection contre la suppression) est désactivée, vous pouvez supprimer le cluster.

Pour supprimer un cluster :

1. Dans le panneau de navigation, choisissez Clusters.
2. Déterminez si le cluster a des instances en examinant la colonne Instances. Avant de pouvoir supprimer un cluster, vous devez supprimer toutes ses instances. Pour plus d'informations, consultez Supprimer une instance Amazon DocumentDB .
3. Selon si votre cluster a des instances, effectuez l'une des étapes suivantes.
   - Si le cluster n'a aucune instance, sélectionnez le bouton situé à gauche du nom du cluster et choisissez Actions. Dans le menu déroulant, choisissez Delete (Supprimer). Renseignez la boîte de dialogue Delete (Supprimer) <nom-cluster>, puis choisissez Delete (Supprimer).

- Si le cluster a une ou plusieurs instances, procédez comme suit :

  a. Dans le panneau de navigation, sélectionnez Instances.

  b. Supprimez toutes les instances du cluster. Lorsque vous supprimez la dernière instance, le cluster est également supprimé. Pour plus d'informations sur la suppression des instances, consultez Supprimer une instance Amazon DocumentDB .

La suppression du cluster prend plusieurs minutes. Pour surveiller le statut du cluster, consultez Surveillance de l'état d'un cluster Amazon DocumentDB.

Using the AWS CLI

Vous ne pouvez pas supprimer un cluster qui a des instances associées. Pour déterminer les instances associées à votre cluster, exécutez la commande `describe-db-clusters` et supprimez toutes les instances du cluster. Ensuite, si nécessaire, désactivez la protection contre la suppression de votre cluster et, enfin, supprimez le cluster.

1. Tout d'abord, supprimer tous les instances de cluster.

   Pour déterminer quelles instances vous devez supprimer, exécutez la commande suivante.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].
[DBClusterIdentifier,DBClusterMembers[*].DBInstanceIdentifier]'
```

   La sortie de cette opération ressemble à ceci (format JSON).

```
[
    [
        "sample-cluster",
        [
            "sample-instance-1",
            "sample-instance-2"
        ]
    ]
]
```

Si le cluster que vous souhaitez supprimer a des instances associées, supprimez-les comme indiqué ci-dessous.

```
aws docdb delete-db-instance \
    --db-instance-identifier sample-instance
```

2. Ensuite, désactivez la protection contre la suppression.

L'utilisation AWS CLI de pour supprimer toutes les instances d'un cluster ne supprime pas le cluster. Vous devez également supprimer le cluster, mais pour ce faire, la protection contre la suppression doit être désactivée.

Pour déterminer si la protection contre la suppression est activée pour le cluster, exécutez la commande suivante.

> **Tip**
>
> Pour connaître l'état de protection contre la suppression de tous vos clusters Amazon DocumentDB, omettez le paramètre. `--db-cluster-identifier`

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].[DBClusterIdentifier,DeletionProtection]'
```

Le résultat de cette opération ressemble à ceci.

```
[
    [
        "sample-cluster",
        "true"
    ]
]
```

Si la protection contre la suppression est activée pour le cluster, modifiez le cluster et désactivez la protection contre la suppression. Pour désactiver la protection contre la suppression sur le cluster, exécutez la commande suivante.

```
aws docdb modify-db-cluster \
    --db-cluster-identifier sample-cluster \
    --no-deletion-protection \
    --apply-immediately
```

3. Enfin, supprimez le cluster.

   Une fois que la protection contre la suppression est désactivée, vous pouvez supprimer le cluster. Pour supprimer un cluster, utilisez l'opération `delete-db-cluster` avec les paramètres suivants.

   - **`--db-cluster-identifier`** : obligatoire. Identifiant du cluster que vous souhaitez supprimer.
   - **`--final-db-snapshot-identifier`**—Facultatif. Si vous voulez un instantané final, vous devez inclure ce paramètre avec un nom pour l'instantané final. Vous devez inclure soit `--final-db-snapshot-identifier` ou `--skip-final-snapshot`.

     Contraintes d'affectation de noms :

     - La longueur est de [1 à 63] lettres, chiffres ou traits d'union.
     - Le premier caractère doit être une lettre.
     - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
     - Doit être unique pour tous les clusters d'Amazon RDS, Amazon Neptune et Amazon DocumentDB par région Compte AWS.
   - **`--skip-final-snapshot`**—Facultatif. Utilisez ce paramètre uniquement si vous ne souhaitez pas prendre un instantané final avant de supprimer votre cluster. Le paramètre par défaut consiste à prendre un instantané final. Vous devez inclure soit `--final-db-snapshot-identifier` ou `--skip-final-snapshot`.

   Le AWS CLI code suivant supprime le cluster `sample-cluster` avec un instantané final. L'opération échoue si des instances sont associées au cluster ou si la protection contre la suppression est activée.

   Example

   Pour Linux, macOS ou Unix :

```
aws docdb delete-db-cluster \
    --db-cluster-identifier sample-cluster \
    --final-db-snapshot-identifier sample-cluster-final-snapshot
```

Pour Windows :

```
aws docdb delete-db-cluster ^
    --db-cluster-identifier sample-cluster ^
    --final-db-snapshot-identifier sample-cluster-final-snapshot
```

Example

Le AWS CLI code suivant supprime le cluster `sample-cluster` sans prendre de capture finale.

Pour Linux, macOS ou Unix :

```
aws docdb delete-db-cluster \
    --db-cluster-identifier sample-cluster \
    --skip-final-snapshot
```

Pour Windows :

```
aws docdb delete-db-cluster ^
    --db-cluster-identifier sample-cluster ^
    --skip-final-snapshot
```

Le résultat de l'opération `delete-db-cluster` est le cluster que vous allez supprimer.

La suppression du cluster prend plusieurs minutes. Pour surveiller le statut du cluster, consultez Surveillance de l'état d'un cluster.

Dimensionnement des clusters Amazon DocumentDB

Amazon DocumentDB vous permet de dimensionner le stockage et le calcul de vos clusters en fonction de vos besoins. Cette section décrit comment vous pouvez utiliser le dimensionnement du stockage, le dimensionnement des instances et le dimensionnement de la lecture pour gérer les performances et le dimensionnement de vos clusters et instances Amazon DocumentDB.

Rubriques

Dimensionnement du stockage

Le stockage Amazon DocumentDB s'adapte automatiquement aux données de votre volume de cluster. À mesure que vos données augmentent, le volume de stockage de votre cluster augmente par incréments de 10 GiB, jusqu'à 128 TiB.

Mise à l'échelle d'instances

Vous pouvez redimensionner votre cluster Amazon DocumentDB selon vos besoins en modifiant la classe d'instance pour chaque instance du cluster. Amazon DocumentDB prend en charge plusieurs classes d'instances optimisées pour Amazon DocumentDB.

Pour plus d’informations, consultez Modification d'une instance Amazon DocumentDB.

Dimensionnement en lecture

Vous pouvez obtenir un dimensionnement de lecture pour votre cluster Amazon DocumentDB en créant jusqu'à 15 répliques Amazon DocumentDB dans le cluster. Chaque réplique Amazon DocumentDB renvoie les mêmes données depuis le volume du cluster avec un décalage de réplication minimal, généralement moins de 100 millisecondes après que l'instance principale a écrit une mise à jour. À mesure que votre trafic de lecture augmente, vous pouvez créer des répliques Amazon DocumentDB supplémentaires et vous y connecter directement pour répartir la charge de lecture de votre cluster. Les répliques Amazon DocumentDB ne doivent pas nécessairement appartenir à la même classe d'instance que l'instance principale.

Pour plus d’informations, consultez Ajouter une instance Amazon DocumentDB à un cluster.

Pour augmenter l'échelle de lecture avec Amazon DocumentDB, nous vous recommandons de vous connecter à votre cluster en tant que jeu de répliques et de distribuer les lectures aux instances de réplication à l'aide des fonctionnalités de préférence de lecture intégrées de votre pilote. Pour en savoir plus, consultez Connexion à Amazon DocumentDB en tant qu'ensemble de réplicas.

Evaluation de l'écriture

Vous pouvez augmenter la capacité d'écriture de votre cluster Amazon DocumentDB en augmentant la taille de l'instance principale de votre cluster. Cette section fournit deux méthodes pour mettre à l'échelle l'instance principale de votre cluster en fonction de vos besoins. La première option vise à minimiser l'impact de l'application, mais implique un plus grand nombre d'étapes. La deuxième option offre plus de simplicité car elle comporte moins d'étapes, mais son impact potentiel sur votre application est plus important.

Selon votre application, vous pouvez choisir l'approche qui vous convient le mieux. Pour plus d'informations sur les tailles d'instance disponibles et les coûts, consultez la page de tarification d'Amazon DocumentDB.

1. Optimisation pour une disponibilité et des performances élevées : si vous vous connectez à votre cluster en mode réplica set (recommandé), vous pouvez utiliser le processus suivant pour minimiser l'impact sur votre application lors du dimensionnement de votre instance principale. Cette méthode minimise l'impact car elle préserve ou accroît la haute disponibilité de votre cluster, et ajoute des cibles de mise à l'échelle en lecture au cluster en tant qu'instances, au lieu de les mettre à jour.

   a. Ajoutez un ou plusieurs réplicas du type d'instance plus important à votre cluster (voir ???). Nous recommandons que tous les réplicas soient du même type d'instance ou d'un type d'instance plus important que celui de l'instance principale. Cela évite une diminution involontaire des performances en écriture que provoquerait le basculement vers un type d'instance plus petit. Pour la plupart des clients, cela signifie doubler temporairement le nombre d'instances dans leur cluster, puis supprimer les réplicas plus petits une fois la mise à l'échelle terminée.

   b. Définissez le niveau de basculement sur la priorité zéro pour tous les nouveaux réplicas, en veillant à ce qu'un réplica du type d'instance plus petit ait la priorité de basculement la plus élevée. Pour plus d'informations, consultez ???.

   c. Lancez un basculement manuel, ce qui entraînera la promotion de l'un des nouveaux réplicas en instance principale. Pour plus d'informations, consultez ???.

   > **Note**
   >
   > Cela entraînera un temps d'arrêt d'environ 30 secondes pour votre cluster. Veuillez prévoir vos opérations en conséquence.

d. Supprimez du cluster tous les réplicas dont le type d'instance est plus petit que celui de votre nouvelle instance principale.

e. Redéfinissez le niveau de basculement de toutes les instances sur la même priorité (généralement, cela signifie de les redéfinir sur 1).

Supposons par exemple que vous avez un cluster qui contient actuellement trois instances `r5.large` (une instance principale et deux réplicas) et que vous souhaitez mettre à l'échelle vers un type d'instance `r5.xlarge`. Pour ce faire, vous devez d'abord ajouter trois instances de réplica `r5.xlarge` à votre cluster, puis définir le niveau de basculement des nouveaux réplicas `r5.xlarge` sur zéro. Vous devez ensuite lancer un basculement manuel (sachant que cela entraînera un temps d'arrêt d'environ 30 secondes de votre application). Une fois le basculement terminé, vous supprimez les trois instances `r5.large` de votre cluster, en laissant le cluster mis à l'échelle avec des instances `r5.xlarge`.

Pour optimiser les coûts, les instances Amazon DocumentDB sont facturées par tranches d'une seconde, avec un minimum de dix minutes après un changement de statut facturable tel que la création, la modification ou la suppression d'une instance. Pour de plus amples informations, veuillez consulter Optimisation des coûts dans la documentation relative aux bonnes pratiques.

2. Optimiser pour la simplicité — Cette approche optimise la simplicité. Il n'étend ni ne contracte le cluster, mais il peut réduire temporairement votre capacité de lecture.

   Il est possible que la modification de la classe d'instance d'une réplique empêche cette instance de traiter les demandes pendant une courte période, de quelques secondes à moins de 30 secondes. Si vous vous connectez à votre cluster en mode réplica set (recommandé), cela réduira votre capacité de lecture d'une réplique (par exemple, à 66 % de capacité dans un cluster à 3 nœuds, ou à 75 % de capacité dans un cluster à 4 nœuds, etc.) pendant l'opération de dimensionnement.

   a. Redimensionnez l'une des instances de réplication de votre cluster. Pour plus d'informations, consultez Gestion de classes d'instance.

   b. Attendez que l'instance soit disponible (voirSurveillance de l'état d'une instance Amazon DocumentDB).

> **Note**
>
> Cela entraînera un temps d'arrêt d'environ 30 secondes pour votre cluster. Veuillez prévoir vos opérations en conséquence.

c. Continuez à exécuter les étapes 1 et 2 jusqu'à ce que toutes les instances de répliques aient été redimensionnées, une par une.

d. Lancez un basculement manuel. Cela fera de l'une des répliques l'instance principale. Pour plus d'informations, consultez Basculement Amazon DocumentDB.

> **Note**
>
> Cela entraînera jusqu'à 30 secondes d'indisponibilité pour votre cluster, mais cela prend souvent moins de temps que cela. Veuillez prévoir vos opérations en conséquence.

e. Redimensionnez l'ancienne instance principale (désormais une réplique).

Clonage d'un volume pour un cluster Amazon DocumentDB

En utilisant le clonage Amazon DocumentDB, vous pouvez créer un nouveau cluster qui utilise le même volume de cluster Amazon DocumentDB et possède les mêmes données que l'original. Le processus est conçu pour être rapide et rentable. Le nouveau cluster avec son volume de données associé est appelé clone. La création d'un clone est plus rapide et plus économe en espace que la copie physique des données à l'aide d'autres techniques telles que la restauration d'instantané.

Amazon DocumentDB prend en charge la création d'un clone provisionné Amazon DocumentDB à partir d'un cluster Amazon DocumentDB provisionné. Lorsque vous créez un clone à l'aide d'une configuration de déploiement différente de celle de la source, le clone est créé à l'aide de la dernière version du moteur Amazon DocumentDB de la source.

Lorsque vous créez des clones à partir de vos clusters Amazon DocumentDB, les clones sont créés dans AWS votre compte, le même compte qui possède le cluster Amazon DocumentDB source.

Rubriques

- Présentation du clonage d'Amazon DocumentDB
- Limites du clonage d'Amazon DocumentDB

- Comment fonctionne le clonage Amazon DocumentDB
- Création d'un clone Amazon DocumentDB

Présentation du clonage d'Amazon DocumentDB

Amazon DocumentDB utilise un copy-on-write protocole pour créer un clone. Ce mécanisme utilise un espace supplémentaire minimal pour créer un clone initial. Lorsque le clone est créé pour la première fois, Amazon DocumentDB conserve une copie unique des données utilisées par le cluster de base de données source et le nouveau cluster Amazon DocumentDB (cloné). L'espace de stockage supplémentaire est alloué uniquement lorsque des modifications sont apportées aux données (sur le volume de stockage Amazon DocumentDB) par le cluster Amazon DocumentDB source ou le clone du cluster Amazon DocumentDB. Pour en savoir plus sur le copy-on-write protocole, voirComment fonctionne le clonage Amazon DocumentDB.

Le clonage Amazon DocumentDB est particulièrement utile pour configurer rapidement des environnements de test à l'aide de vos données de production, sans risquer de les corrompre. Vous pouvez utiliser des clones pour de nombreux types d'applications, telles que les suivantes :

- Expérimentez des changements potentiels (par exemple, des changements de schémas et de groupes de paramètres) pour évaluer tous les impacts.
- Exécutez des opérations imposant une charge de travail élevée, telles que l'exportation de données ou l'exécution de requêtes analytiques sur le clone.
- Créez une copie de votre cluster de base de données de production à des fins de développement, de test ou autres.

Vous pouvez créer plusieurs clones à partir du même cluster Amazon DocumentDB. Vous pouvez également créer plusieurs clones à partir d'un autre clone.

Après avoir créé un clone Amazon DocumentDB, vous pouvez configurer les instances Amazon DocumentDB différemment du cluster Amazon DocumentDB source. Par exemple, il se peut que vous n'ayez pas besoin d'un clone à des fins de développement pour répondre aux mêmes exigences de haute disponibilité que le cluster Amazon DocumentDB de production source. Dans ce cas, vous pouvez configurer le clone avec une seule instance Amazon DocumentDB plutôt qu'avec les multiples instances de base de données utilisées par le cluster Amazon DocumentDB.

Lorsque vous avez fini d'utiliser le clone à des fins de test, de développement ou autres, vous pouvez le supprimer.

Limites du clonage d'Amazon DocumentDB

Amazon DocumentDB ; le clonage présente actuellement les limites suivantes :

- Vous pouvez créer autant de clones que vous le souhaitez, jusqu'au nombre maximal de clusters de bases de données autorisés dans la Région AWS. Toutefois, lorsque vous avez créé 15 clones, le 16ème est une copie intégrale. L'opération de clonage agit comme une point-in-time restauration.
- Vous ne pouvez pas créer de clone dans une AWS région différente de celle du cluster Amazon DocumentDB source.
- Vous ne pouvez pas créer de clone à partir d'un cluster Amazon DocumentDB dépourvu d'instances de base de données. Vous ne pouvez cloner que des clusters Amazon DocumentDB dotés d'au moins une instance de base de données.
- Vous pouvez créer un clone dans un cloud privé virtuel (VPC) différent de celui du cluster Amazon DocumentDB. Cependant, les sous-réseaux des VPC doivent mapper aux mêmes zones de disponibilité.

Comment fonctionne le clonage Amazon DocumentDB

Le clonage Amazon DocumentDB fonctionne au niveau de la couche de stockage d'un cluster Amazon DocumentDB. Il utilise un copy-on-writeprotocole à la fois rapide et peu encombrant en termes de support durable sous-jacent supportant le volume de stockage Amazon DocumentDB. Vous pouvez en savoir plus sur les volumes de cluster Amazon DocumentDB dans. Gestion des clusters Amazon DocumentDB

Rubriques

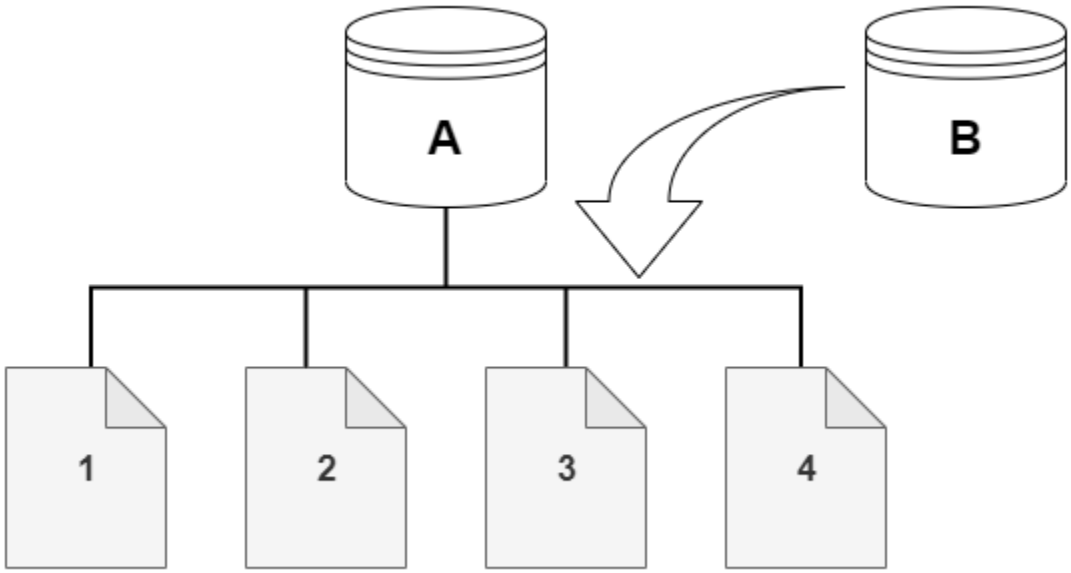
- Comprendre le copy-on-write protocole
- Suppression d'un volume de cluster source

Comprendre le copy-on-write protocole

Un cluster Amazon DocumentDB stocke les données dans des pages du volume de stockage Amazon DocumentDB sous-jacent.

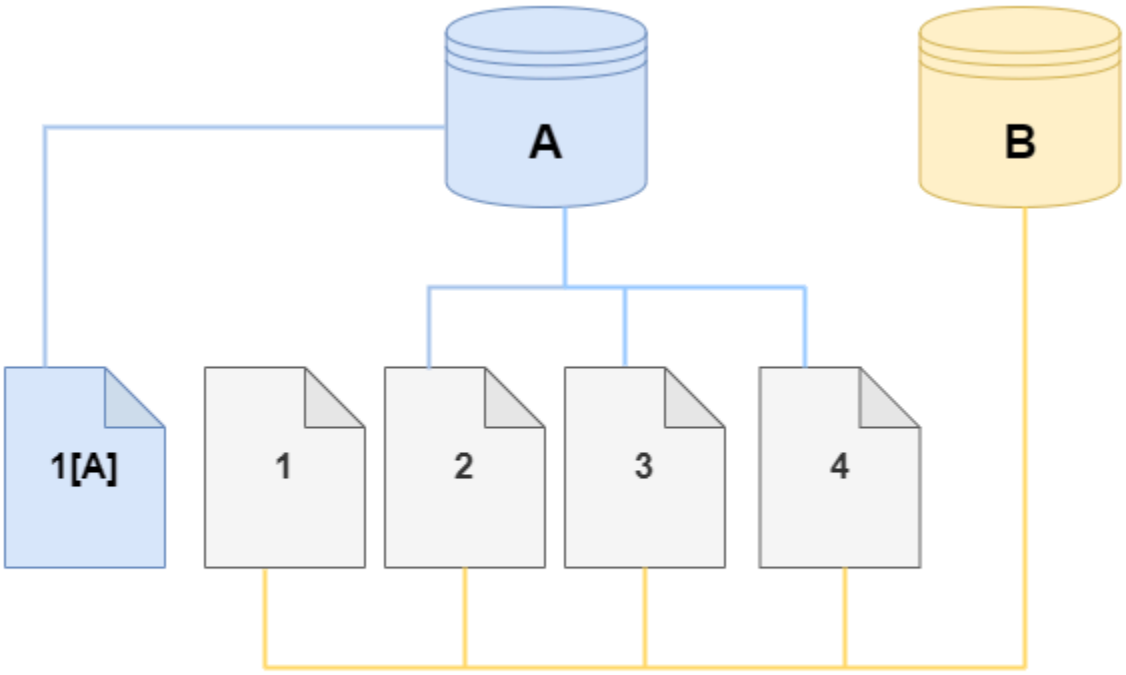
Par exemple, dans le schéma suivant, vous pouvez trouver un cluster Amazon DocumentDB (A) comportant quatre pages de données, 1, 2, 3 et 4. Imaginez qu'un clone, B, soit créé à partir du

cluster Amazon DocumentDB. Lors de la création du clone, aucune donnée n'est copiée. Le clone pointe plutôt vers le même ensemble de pages que le cluster Amazon DocumentDB source.

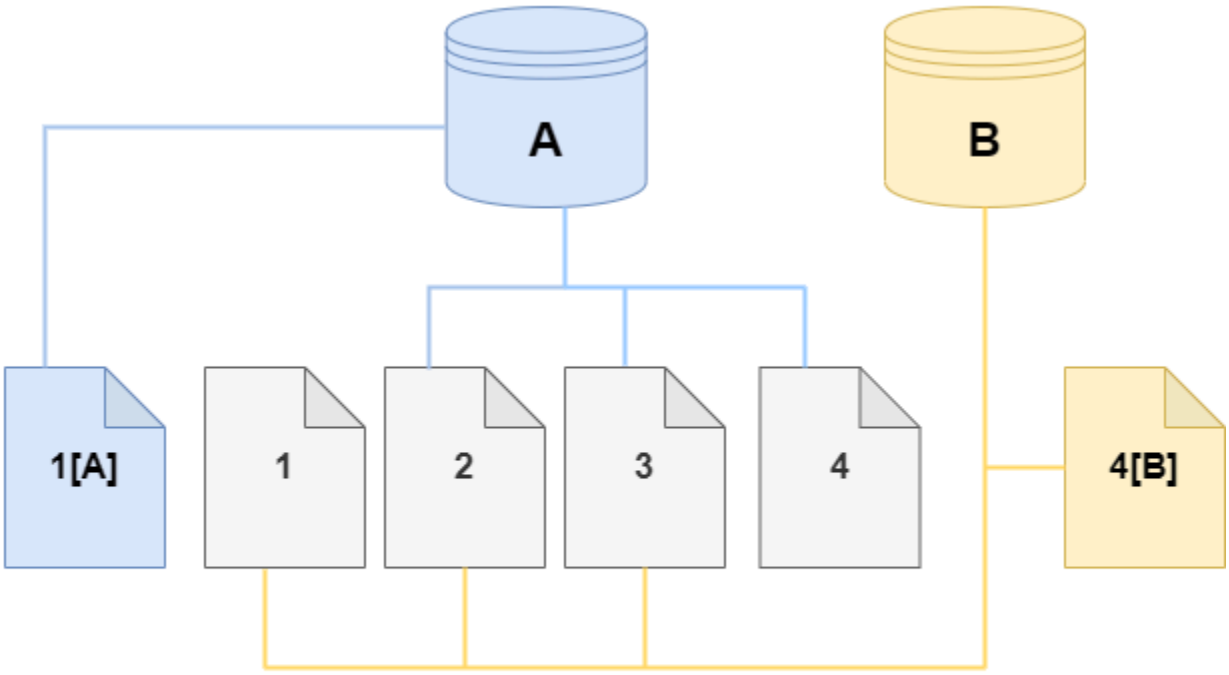


Lors de la création du clone, aucun stockage supplémentaire n'est généralement nécessaire. Le copy-on-write protocole utilise le même segment sur le support de stockage physique que le segment source. Un stockage supplémentaire n'est requis que si la capacité du segment source n'est pas suffisante pour le segment de clone entier. Dans ce cas, le segment source est copié sur un autre périphérique physique.

Dans les diagrammes suivants, vous pouvez trouver un exemple du copy-on-write protocole en action utilisant le même cluster A et son clone, B, comme indiqué ci-dessus. Supposons que vous apportiez une modification à votre cluster Amazon DocumentDB (A) qui entraîne une modification des données contenues sur la page 1. Au lieu d'écrire sur la page 1 d'origine, Amazon DocumentDB crée une nouvelle page 1 [A]. Le volume du cluster Amazon DocumentDB pour le cluster (A) pointe désormais vers les pages 1 [A], 2, 3 et 4, tandis que le clone (B) fait toujours référence aux pages d'origine.



Sur le clone, une modification est apportée à la page 4 sur le volume de stockage. Au lieu d'écrire sur la page 4 d'origine, Amazon DocumentDB crée une nouvelle page, 4 [B]. Le clone pointe maintenant vers les pages 1, 2, 3 et 4[B], tandis que le cluster (A) continue de pointer vers les pages 1[A], 2, 3 et 4.



Au fur et à mesure que de nouvelles modifications se produisent au fil du temps, à la fois dans le volume du cluster Amazon DocumentDB source et dans le clone, davantage de stockage est nécessaire pour capturer et stocker les modifications.

Suppression d'un volume de cluster source

Lorsque vous supprimez un volume de cluster source auquel un ou plusieurs clones sont associés, ceux-ci ne sont pas affectés. Les clones continuent de pointer vers les pages qui étaient précédemment la propriété du volume de cluster source.

Création d'un clone Amazon DocumentDB

Vous pouvez créer un clone dans le même AWS compte que le cluster Amazon DocumentDB source. Pour ce faire, vous pouvez utiliser le AWS Management Console ou les AWS CLI et les procédures suivantes.

En utilisant le clonage Amazon DocumentDB, vous pouvez créer un clone de cluster Amazon DocumentDB provisionné à partir d'un cluster Amazon DocumentDB provisionné.

Using the AWS Management Console

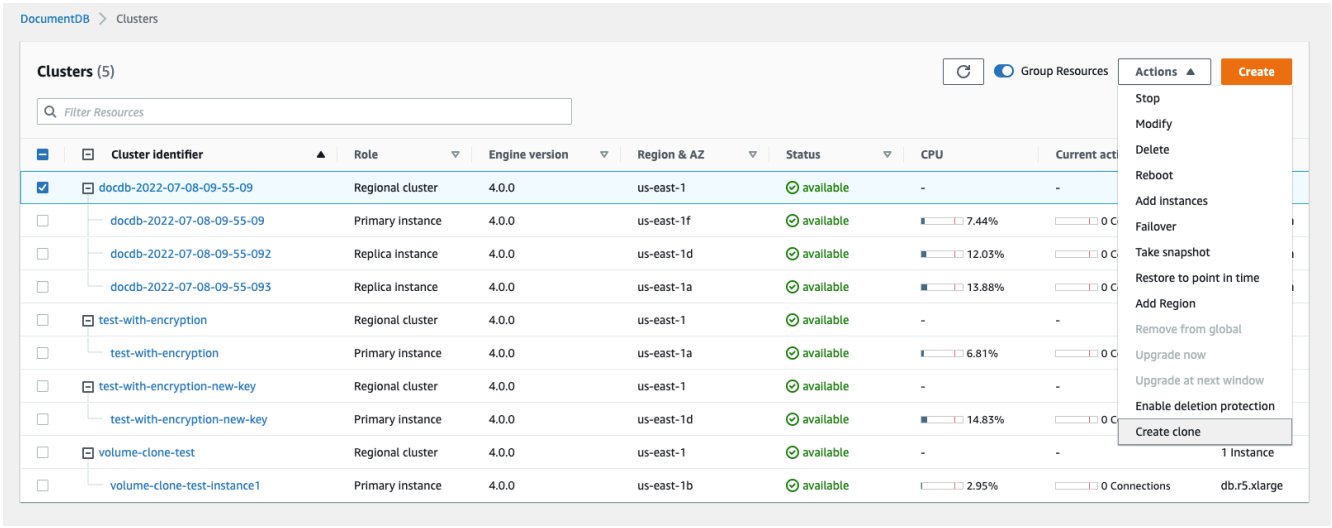
La procédure suivante décrit comment cloner un cluster Amazon DocumentDB à l'aide du. AWS Management Console

Création d'un clone à l'aide des AWS Management Console résultats dans un cluster Amazon DocumentDB avec une instance Amazon DocumentDB.

Ces instructions s'appliquent aux clusters de base de données appartenant au même AWS compte qui crée le clone. Le cluster de base de données doit appartenir au même AWS compte, car le clonage entre comptes n'est pas pris en charge dans Amazon DocumentDB.

Pour créer un clone d'un cluster de base de données appartenant à votre AWS compte à l'aide du AWS Management Console

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.
3. Choisissez votre cluster Amazon DocumentDB dans la liste, et pour Actions, choisissez Create clone.

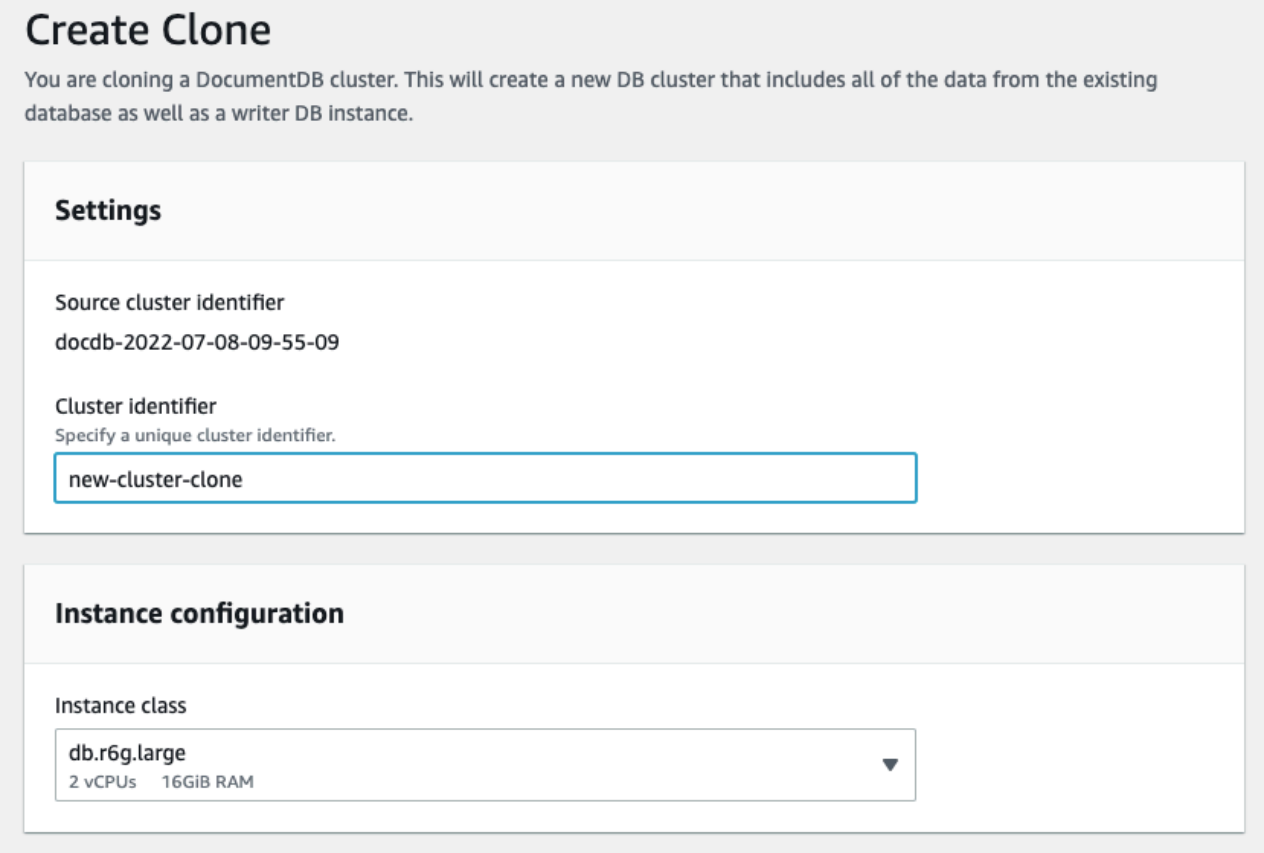


La page Créer un clone s'ouvre, où vous pouvez configurer un identifiant de cluster et une classe d'instance, ainsi que d'autres options pour le clone de cluster Amazon DocumentDB.

4. Dans la section Settings (Paramètres), procédez comme suit :

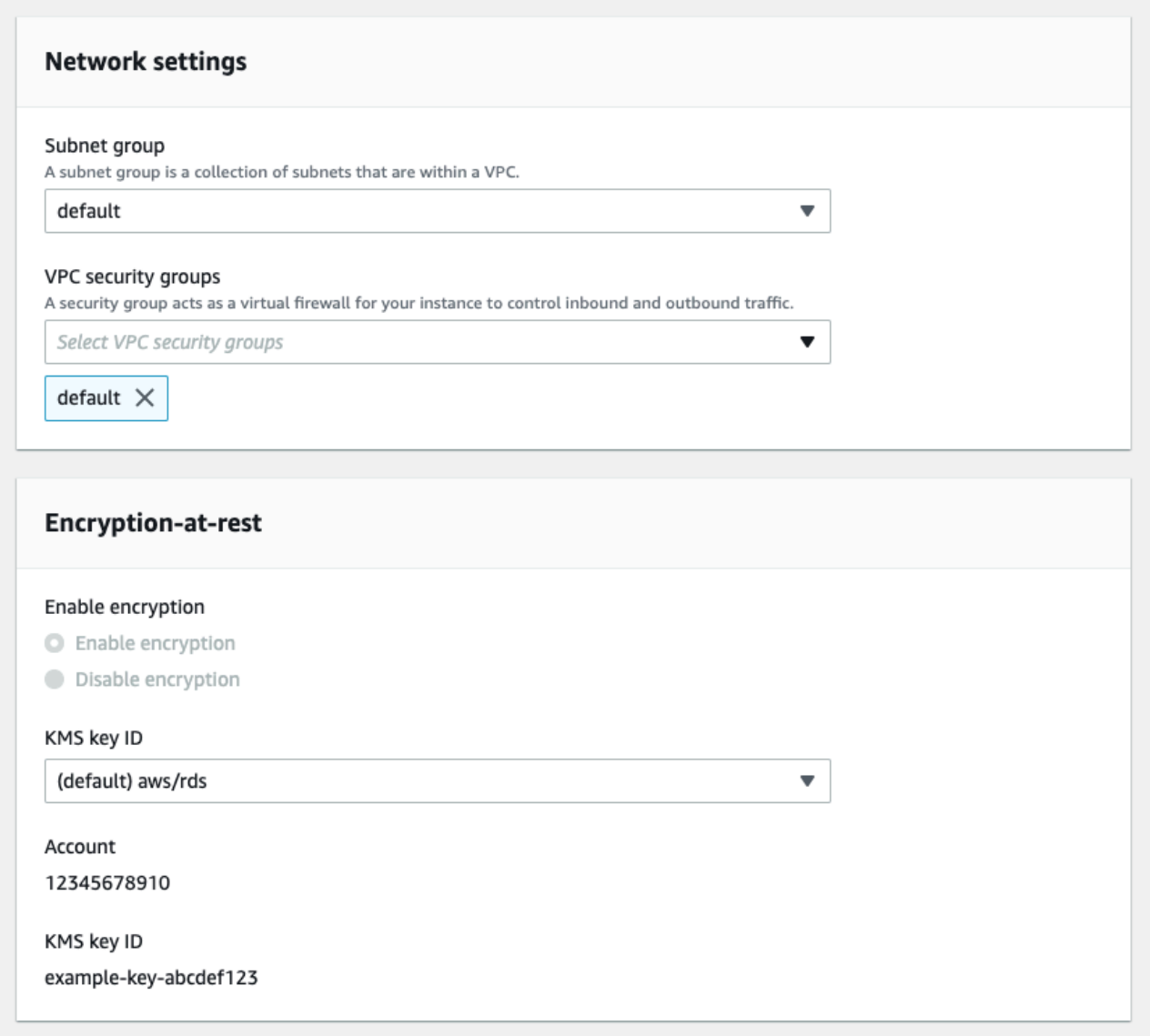
   a. Dans le champ Identifiant du cluster, entrez le nom que vous souhaitez attribuer à votre cluster Amazon DocumentDB cloné.

b. Pour la configuration de l'instance, sélectionnez une classe d'instance appropriée pour votre cluster Amazon DocumentDB cloné.

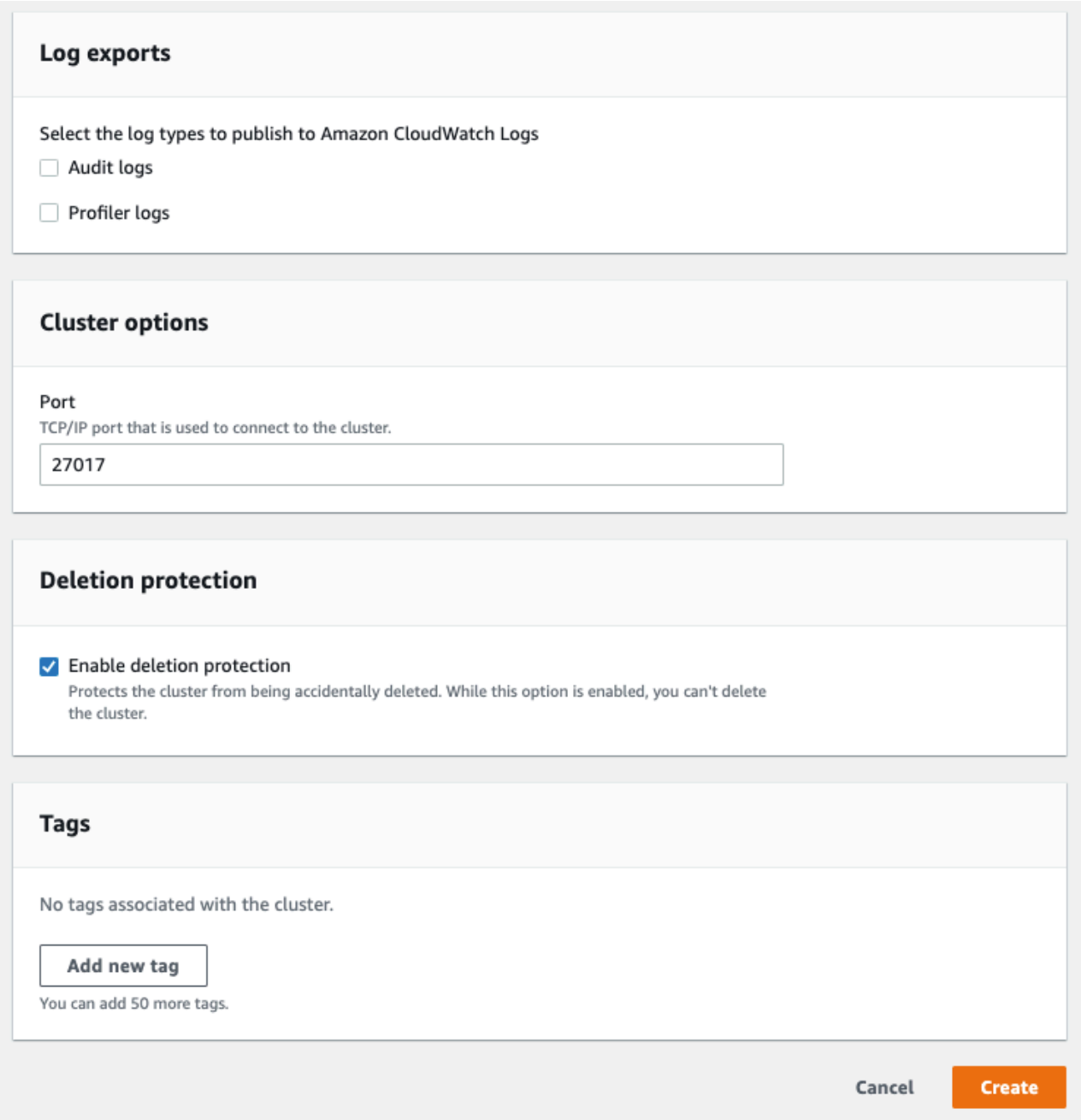


c. Pour les paramètres réseau, choisissez un groupe de sous-réseaux pour votre cas d'utilisation et les groupes de sécurité VPC associés.

d. Pour E ncryption-at-rest, si le chiffrement est activé sur le cluster source (le cluster cloné), le chiffrement doit également être activé sur le cluster cloné. Si ce scénario est vrai, les options Activer le chiffrement sont grisées (désactivées) mais l'option Activer le chiffrement est sélectionnée. À l'inverse, si le chiffrement n'est pas activé sur le cluster source, les options Activer le chiffrement sont disponibles et vous pouvez choisir d'activer ou de désactiver le chiffrement.



e. Complétez la nouvelle configuration du clone de cluster en sélectionnant le type de journaux à exporter (facultatif), en saisissant un port spécifique utilisé pour se connecter au cluster et en activant la protection contre la suppression accidentelle du cluster (activée par défaut).



f. Terminez de saisir tous les paramètres de votre clone de cluster Amazon DocumentDB. Pour en savoir plus sur les paramètres du cluster et de l'instance Amazon DocumentDB, consultez. Gestion des clusters Amazon DocumentDB

5. Choisissez Create clone pour lancer le clone Amazon DocumentDB du cluster Amazon DocumentDB de votre choix.

Lorsque le clone est créé, il est répertorié avec vos autres clusters Amazon DocumentDB dans la section Bases de données de la console et affiche son état actuel. Votre clone est prêt à être utilisé quand son état est Disponible.

Using the AWS CLI

L'utilisation du AWS CLI pour cloner votre cluster Amazon DocumentDB implique quelques étapes.

La `restore-db-cluster-to-point-in-time` AWS CLI commande que vous utilisez génère un cluster Amazon DocumentDB vide avec 0 instance Amazon DocumentDB. En d'autres termes, la commande restaure uniquement le cluster Amazon DocumentDB, et non les instances de base de données de ce cluster. Vous faites cela séparément une fois le clone disponible. Les deux étapes du processus sont les suivantes :

1. Créez le clone à l'aide de la commande restore-db-cluster-to- point-in-time CLI. Les paramètres que vous utilisez avec cette commande contrôlent le type de capacité et d'autres détails du cluster Amazon DocumentDB vide (clone) en cours de création.
2. Créez l'instance Amazon DocumentDB pour le clone à l'aide de la commande create-db-instanceCLI pour recréer l'instance Amazon DocumentDB dans le cluster Amazon DocumentDB restauré.

Les commandes suivantes supposent que votre AWS région AWS CLI est configurée par défaut. Cette approche vous évite de passer le nom de `--region` dans chaque commande. Pour plus d'informations, consultez Configuration de l' AWS CLI. Vous pouvez également spécifier la `--region` dans chacune des commandes de la CLI qui suivent.

Création du clone

Les paramètres spécifiques que vous passez à la commande de la CLI `restore-db-cluster-to-point-in-time` varient. Ce que vous transmettez dépend du type de clone que vous souhaitez créer.

Utilisez la procédure suivante pour créer un clone Amazon DocumentDB provisionné à partir d'un cluster Amazon DocumentDB provisionné.

Pour créer un clone du même mode de moteur que le cluster Amazon DocumentDB source

- Utilisez la commande de la CLI `restore-db-cluster-to-point-in-time` et spécifiez les valeurs des paramètres suivants :

- `--db-cluster-identifier` – Choisissez un nom explicite pour votre clone. Vous nommez le clone lorsque vous utilisez la commande restore-db-cluster-to- point-in-time CLI.
- `--restore-type` – Utilisez la commande `copy-on-write` pour créer un clone du cluster de base de données source. Sans ce paramètre, le cluster Amazon DocumentDB `restore-db-cluster-to-point-in-time` restaure plutôt que de créer un clone. La valeur par défaut pour `restore-type` est`full-copy`.
- `--source-db-cluster-identifier`— Utilisez le nom du cluster Amazon DocumentDB source que vous souhaitez cloner.
- `--use-latest-restorable-time` – Cette valeur pointe vers les données de volume restaurables les plus récentes pour le clone. Ce paramètre est obligatoire car`restore-type copy-on-write`, cependant, vous ne pouvez pas utiliser le `restore-to-time parameter` avec.

L'exemple suivant crée un clone nommé `my-clone` à partir d'un cluster nommé `my-source-cluster`.

Pour Linux, macOS ou Unix :

```
aws docdb restore-db-cluster-to-point-in-time \
    --source-db-cluster-identifier my-source-cluster \
    --db-cluster-identifier my-clone \
    --restore-type copy-on-write \
    --use-latest-restorable-time
```

Pour Windows :

```
aws docdb restore-db-cluster-to-point-in-time ^
    --source-db-cluster-identifier my-source-cluster ^
    --db-cluster-identifier my-clone ^
    --restore-type copy-on-write ^
    --use-latest-restorable-time
```

La commande renvoie l'objet JSON contenant les détails du clone. Vérifiez que votre cluster de base de données cloné est disponible avant d'essayer de créer l'instance de base de données pour votre clone. Pour plus d'informations, consultez la section Vérification de l'état et obtention des informations relatives au clone ci-dessous :

Vérifier le statut et obtenir les détails du clone

Vous pouvez utiliser la commande suivante pour vérifier l'état de votre cluster de base de données vide nouvellement créé.

```
$ aws docdb describe-db-clusters --db-cluster-identifier my-clone --query '*[].
[Status]' --output text
```

Vous pouvez également obtenir le statut et les autres valeurs dont vous avez besoin pour créer l'instance de base de données pour votre clone en utilisant la AWS CLI requête suivante :

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-clusters --db-cluster-identifier my-clone \
  --query '*[].{Status:Status,Engine:Engine,EngineVersion:EngineVersion}'
```

Pour Windows :

```
aws docdb describe-db-clusters --db-cluster-identifier my-clone ^
  --query "*[].{Status:Status,Engine:Engine,EngineVersion:EngineVersion}"
```

Cette requête retourne une sortie similaire à la suivante.

```
[
  {
        "Status": "available",
        "Engine": "docdb",
        "EngineVersion": "4.0.0",
    }
]
```

Création de l'instance Amazon DocumentDB pour votre clone

Utilisez la commande create-db-instanceCLI pour créer l'instance de base de données pour votre clone.

Le `--db-instance-class` paramètre est utilisé uniquement pour les clusters Amazon DocumentDB provisionnés.

Pour Linux, macOS ou Unix :

```
aws docdb create-db-instance \
```

```
    --db-instance-identifier my-new-db \
    --db-cluster-identifier my-clone \
    --db-instance-class  db.r5.4xlarge \
    --engine docdb
```

Pour Windows :

```
aws docdb create-db-instance ^
    --db-instance-identifier my-new-db ^
    --db-cluster-identifier my-clone ^
    --db-instance-class  db.r5.4xlarge ^
    --engine docdb
```

Paramètres à utiliser pour le clonage

Le tableau suivant récapitule les différents paramètres utilisés pour cloner des `restore-db-cluster-to-point-in-time` clusters Amazon DocumentDB.

Paramètre	Description
--source-db-cluster-identifier	Utilisez le nom du cluster Amazon DocumentDB source que vous souhaitez cloner.
--db-cluster-identifier	Choisissez un nom explicite pour votre clone. Vous nommez votre clone à l'aide de la commande `restore-db-cluster-to-point-in-time` . Ensuite, vous passez ce nom à la commande `create-db-instance` .
—type de restauration	Spécifiez `copy-on-write` comme `--restore-type` pour créer un clone du cluster de base de données source plutôt que de restaurer le cluster Amazon DocumentDB source.
--use-latest-restorable-time	Cette valeur pointe vers les données de volume restaurables les plus récentes pour le clone.

Comprendre la tolérance aux pannes des clusters Amazon DocumentDB

Les clusters Amazon DocumentDB sont par conception tolérants aux pannes. Le volume de chaque cluster couvre plusieurs zones de disponibilité en une seule Région AWS, et chaque zone de

disponibilité contient une copie des données de volume du cluster. Cette fonctionnalité signifie que votre cluster peut tolérer une défaillance d'une zone de disponibilité sans perte de données et uniquement une brève interruption de service.

En cas de défaillance de l'instance principale d'un cluster, Amazon DocumentDB effectue automatiquement un basculement vers une nouvelle instance principale de deux manières :

- En promouvant une réplique Amazon DocumentDB existante vers la nouvelle instance principale choisie en fonction du paramètre de niveau de promotion de chaque réplique, puis en créant une instance de remplacement pour l'ancienne instance principale. Le basculement vers l'instance de réplique prend généralement moins de 30 secondes. Les opérations de lecture et d'écriture peuvent être brièvement interrompues pendant cette période. Pour augmenter la disponibilité de votre cluster, nous vous recommandons de créer au moins une ou plusieurs répliques Amazon DocumentDB dans au moins deux zones de disponibilité différentes.
- Par la création d'une nouvelle instance principale. Cela ne se produit que si vous ne disposez pas d'une instance de réplication dans votre cluster et cela peut prendre quelques minutes.

Si le cluster possède une ou plusieurs répliques Amazon DocumentDB, une réplique Amazon DocumentDB est promue en instance principale en cas de défaillance. Un événement d'échec se traduit par une brève interruption, pendant laquelle les opérations de lecture et d'écriture échouent avec une exception. Cependant, le service est généralement restauré en moins de 120 secondes, et souvent en moins de 60 secondes. Pour augmenter la disponibilité de votre cluster, nous vous recommandons de créer au moins une ou plusieurs répliques Amazon DocumentDB dans au moins deux zones de disponibilité différentes.

Vous pouvez personnaliser l'ordre dans lequel vos répliques Amazon DocumentDB sont promues vers l'instance principale après une panne en attribuant une priorité à chaque réplique. Les priorités s'étendent de la valeur 0 pour la plus haute priorité à la valeur 15 pour la plus basse priorité. En cas de défaillance de l'instance principale, la réplique Amazon DocumentDB ayant la priorité la plus élevée est promue vers la nouvelle instance principale. Vous pouvez modifier la priorité d'une réplique Amazon DocumentDB à tout moment. La modification de la priorité ne déclenche pas un basculement. Vous pouvez utiliser l'opération `modify-db-instance` avec le paramètre `--promotion-tier`. Pour en savoir plus sur la personnalisation de la priorité de basculement d'une instance, consultez Basculement Amazon DocumentDB.

Plusieurs répliques Amazon DocumentDB peuvent partager la même priorité, ce qui se traduit par des niveaux de promotion. Si deux répliques Amazon DocumentDB ou plus partagent la même priorité, la réplique la plus grande est promue en tant que réplique principale. Si deux répliques

Amazon DocumentDB ou plus partagent la même priorité et la même taille, une réplique arbitraire appartenant au même niveau de promotion est promue.

Si le cluster ne contient aucune réplique Amazon DocumentDB, l'instance principale est recréée lors d'un événement de défaillance. Un événement d'échec se traduit par une interruption, pendant laquelle les opérations de lecture et d'écriture échouent avec une exception. Le service est rétabli quand la nouvelle instance principale est créée, ce qui prend généralement moins de 10 minutes. La promotion d'une réplique Amazon DocumentDB vers l'instance principale est beaucoup plus rapide que la création d'une nouvelle instance principale.

Gestion des instances Amazon DocumentDB

Les rubriques suivantes fournissent des informations qui vous aideront à gérer vos instances Amazon DocumentDB. Elles incluent des détails sur les classes et les statuts des instances, et sur la façon de créer, supprimer et modifier une instance.

Rubriques

- Gestion de classes d'instance
- Identification du statut d'une instance
- Cycle de vie des instances Amazon DocumentDB

Gestion de classes d'instance

La classe d'instance détermine le calcul et la capacité de mémoire d'une instance Amazon DocumentDB (compatible avec MongoDB). La classe d'instance dont vous avez besoin varie selon vos exigences en mémoire et en puissance de traitement.

Amazon DocumentDB prend en charge les familles de classes d'instances R4, R5, R6G, T3 et T4G. Il s'agit de classes d'instance de la génération actuelle optimisées pour les applications exigeantes en mémoire. Pour les spécifications sur ces classes, consultez Spécifications de la classe d'instance.

Rubriques

- Déterminer une classe d'instance
- Modification de la classe d'une instance
- Classes d'instances prises en charge par région
- Spécifications de la classe d'instance

Déterminer une classe d'instance

Pour déterminer la classe d'une instance, vous pouvez utiliser l'`describe-db-instances` AWS CLI opération AWS Management Console ou.

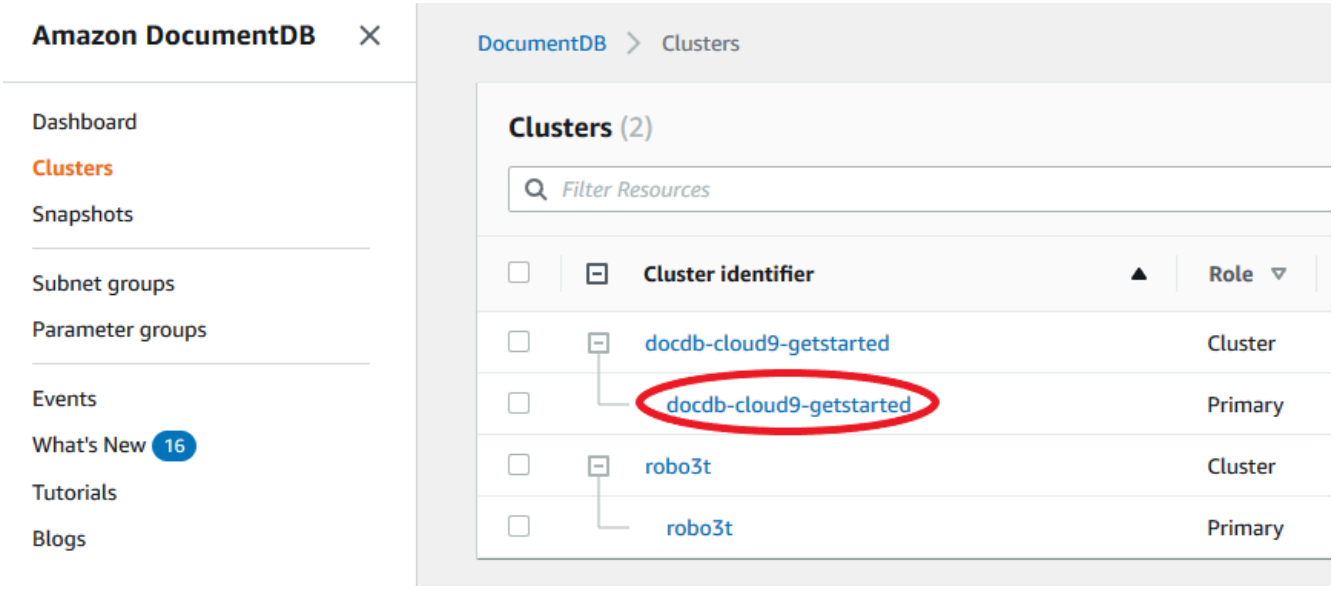
Using the AWS Management Console

Pour déterminer la classe d'instance pour les instances de votre cluster, effectuez les étapes suivantes dans la console.

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le volet de navigation, choisissez Clusters pour trouver l'instance qui vous intéresse.

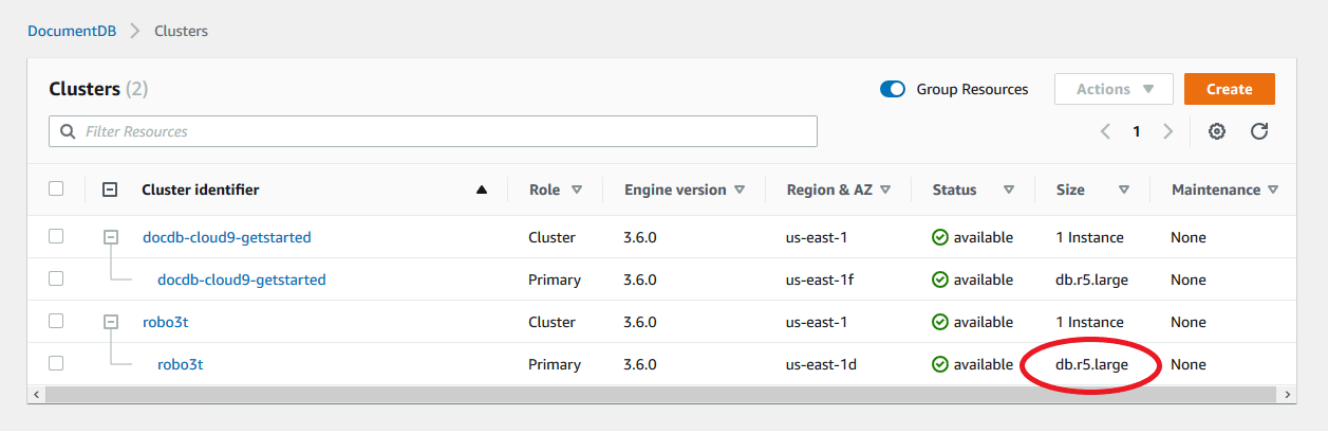
   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans la boîte de navigation Clusters, vous verrez la colonne Cluster Identifier. Vos instances sont répertoriées sous des clusters, comme dans la capture d'écran ci-dessous.



4. Dans la liste des instances, développez le cluster pour trouver les instances qui vous intéressent. Trouvez l'instance que vous souhaitez. Regardez ensuite la colonne Size de la ligne de l'instance pour voir sa classe d'instance.

   Dans l'image suivante, la classe d'instance pour `robo3t` est `db.r5.4xlarge`.



Using the AWS CLI

Pour déterminer la classe d'une instance à l'aide de AWS CLI, utilisez l'`describe-db-instances`opération avec les paramètres suivants.

- **`--db-instance-identifier`**— Facultatif. Spécifie l'instance pour laquelle vous voulez déterminer la classe. Si cet élément n'est pas spécifié, `describe-db-instances` renvoie une description de 100 de vos instances au plus.
- **`--query`**— Facultatif. Spécifie les membres de l'instance à inclure dans les résultats. Si ce paramètre n'est pas spécifié, tous les membres de l'instance sont renvoyés.

Example

L'exemple suivant permet de trouver le nom et la classe de l'instance`sample-instance-1`.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' \
    --db-instance-identifier sample-instance-1
```

Pour Windows :

```
aws docdb describe-db-instances ^
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' ^
    --db-instance-identifier sample-instance-1
```

Le résultat de cette opération ressemble à ceci.

```
[
    [
        "sample-instance-1",
        "db.r5.large"
    ]
```

Example

L'exemple suivant trouve le nom et la classe de l'instance pour un maximum de 100 instances Amazon DocumentDB.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' \
    --filter Name=engine,Values=docdb
```

Pour Windows :

```
aws docdb describe-db-instances ^
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' ^
    --filter Name=engine,Values=docdb
```

Le résultat de cette opération ressemble à ceci.

```
[
    [
        "sample-instance-1",
        "db.r5.large"
    ],
    [
        "sample-instance-2",
        "db.r5.large"
    ],
    [
```

```
        "sample-instance-3",
        "db.r5.4xlarge"
    ],
    [
        "sample-instance-4",
        "db.r5.4xlarge"
    ]
]
```

Pour plus d'informations, consultez Décrire les instances Amazon DocumentDB.

Modification de la classe d'une instance

Vous pouvez modifier la classe d'instance de votre instance à l'aide du AWS Management Console ou du AWS CLI. Pour plus d'informations, consultez Modification d'une instance Amazon DocumentDB.

Classes d'instances prises en charge par région

Amazon DocumentDB prend en charge les classes d'instance suivantes :

- R6G—Dernière génération d'instances optimisées pour la mémoire alimentées par des processeurs AWS Graviton2 basés sur ARM qui offrent des performances jusqu'à 30 % supérieures à celles des instances R5 à un coût 5 % inférieur.
- R5—Instances optimisées pour la mémoire qui offrent des performances jusqu'à 100 % supérieures à celles des instances R4 pour le même coût d'instance.
- R4—Génération précédente d'instances optimisées pour la mémoire.
- T4G—Type d'instance généraliste à faible coût et à faible coût de dernière génération alimenté par des processeurs AWS Graviton2 basés sur ARM qui fournit un niveau de performance CPU de référence, offrant un rapport prix/performances jusqu'à 35 % supérieur à celui des instances T3 et idéal pour exécuter des applications avec une utilisation modérée du processeur qui connaissent des pics d'utilisation temporaires.
- T3—Type d'instance généraliste à faible coût qui fournit un niveau de performance du processeur de référence avec la possibilité d'augmenter l'utilisation du processeur à tout moment et aussi longtemps que nécessaire.

Pour obtenir des spécifications détaillées sur les classes d'instances, veuillez consulter Spécifications de la classe d'instance.

Une classe d'instance particulière peut être prise en charge ou non dans une région donnée. Le tableau suivant indique quelles classes d'instances sont prises en charge par Amazon DocumentDB dans chaque région.

Classes d'instances prises en charge par région

Région	R6G	R5	R4	T4G	T3
USA Est (Ohio)	Pris en charge	Pris en charge	Pris en charge	Pris en charge	Pris en charge
USA Est (Virginie du Nord)	Pris en charge	Pris en charge	Pris en charge	Pris en charge	Pris en charge
USA Ouest (Oregon)	Pris en charge	Pris en charge	Pris en charge	Pris en charge	Pris en charge
Amérique du Sud (São Paulo)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Asie-Pacifique (Hong Kong)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Asie-Pacifique (Hyderabad)		Pris en charge			Pris en charge
Asie-Pacifique (Mumbai)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Asie-Pacifique (Séoul)	Pris en charge	Pris en charge		Pris en charge	Pris en charge

Région	R6G	R5	R4	T4G	T3
Asie-Pacifique (Sydney)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Asie-Pacifique (Singapour)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Asie-Pacifique (Tokyo)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Canada (Centre)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Europe (Francfort)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Europe (Irlande)	Pris en charge	Pris en charge	Pris en charge	Pris en charge	Pris en charge
Europe (Londres)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Europe (Milan)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Europe (Paris)	Pris en charge	Pris en charge		Pris en charge	Pris en charge

Région	R6G	R5	R4	T4G	T3
Moyen-Orient (EAU)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Région Chine (Beijing)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
Chine (Ningxia)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
AWS GovCloud (US-Ouest)	Pris en charge	Pris en charge		Pris en charge	Pris en charge
AWS GovCloud (USA Est)	Pris en charge	Pris en charge		Pris en charge	Pris en charge

Spécifications de la classe d'instance

Le tableau suivant fournit des informations détaillées sur les classes d'instance Amazon DocumentDB. Vous trouverez des explications pour chaque colonne de tableau sous ce dernier.

Classes d'instance Amazon DocumentDB prises en charge

Classe d'instance	vCPU[1]	Mémoire (GiB) 2	Température maximale de stockage (GiB) 3	Bande passante max (Mo/s)[4]	Performances réseau[5]	Moteurs de soutien [6]
R6G — Classe d'instance optimisée pour la mémoire de génération actuelle basée sur Graviton2						
`db.r6g.large`	2	16	32	Jusqu'à 4 750	Jusqu'à 10 Gbit/s	4.0.0 et 5.0.0

Classe d'instance	vCPU[1]	Mémoire (GiB) 2	Température maximale de stockage (GiB) 3	Bande passante max (Mo/s)[4]	Performances réseau[5]	Moteurs de soutien [6]
db.r6g.xlarge	4	32	63	Jusqu'à 4 750	Jusqu'à 10 Gbit/s	4.0.0 et 5.0.0
db.r6g.2xlarge	8	64	126	Jusqu'à 4 750	Jusqu'à 10 Gbit/s	4.0.0 et 5.0.0
db.r6g.4xlarge	16	128	252	4 750	Jusqu'à 10 Gbit/s	4.0.0 et 5.0.0
db.r6g.8xlarge	32	256	504	9 000	12 Gb/s	4.0.0 et 5.0.0
db.r6g.12xlarge	48	384	756	13 500	20 Gbit/s	4.0.0 et 5.0.0
db.r6g.16xlarge	64	512	1008	19 000	25 Gb/s	4.0.0 et 5.0.0
R5 — Classe d'instance optimisée pour la mémoire de la génération précédente						
db.r5.large	2	16	31	Jusqu'à 3 500	Jusqu'à 10 Gbit/s	3.6.0, 4.0.0 et 5.0.0
db.r5.xlarge	4	32	62	Jusqu'à 3 500	Jusqu'à 10 Gbit/s	3.6.0, 4.0.0 et 5.0.0
db.r5.2xlarge	8	64	124	Jusqu'à 3 500	Jusqu'à 10 Gbit/s	3.6.0, 4.0.0 et 5.0.0
db.r5.4xlarge	16	128	249	3 500	Jusqu'à 10 Gbit/s	3.6.0, 4.0.0 et 5.0.0
db.r5.8xlarge	32	256	504	6 800	10 Gbit/s	3.6.0, 4.0.0 et 5.0.0

Classe d'instance	vCPU[1]	Mémoire (GiB)[2]	Température maximale de stockage (GiB)[3]	Bande passante max (Mo/s)[4]	Performances réseau[5]	Moteurs de soutien[6]
db.r5.12xlarge	48	384	748	7 000	10 Gbit/s	3.6.0, 4.0.0 et 5.0.0
db.r5.16xlarge	64	512	1008	13 600	20 Gb/s	3.6.0, 4.0.0 et 5.0.0
db.r5.24xlarge	96	768	1 500	14 000	25 Gbit/s	3.6.0, 4.0.0 et 5.0.0
R4 — Classe d'instance optimisée pour la mémoire de la génération précédente						
db.r4.large	2	15,25	30	437	Jusqu'à 10 Gbit/s	3.6.0 uniquement
db.r4.xlarge	4	30,5	60	875	Jusqu'à 10 Gbit/s	3.6.0 uniquement
db.r4.2xlarge	8	61	120	875	Jusqu'à 10 Gbit/s	3.6.0 uniquement
db.r4.4xlarge	16	122	240	875	Jusqu'à 10 Gbit/s	3.6.0 uniquement
db.r4.8xlarge	32	244	480	875	10 Gbit/s	3.6.0 uniquement
db.r4.16xlarge	64	488	960	14 000	25 Gbit/s	3.6.0 uniquement

Classe d'instance	vCPU[1]	Mémoire (GiB) 2	Température maximale de stockage (GiB) 3	Bande passante max (Mo/s)[4]	Performances réseau[5]	Moteurs de soutien [6]
T4G — Classes d'instances de performances éclatantes de dernière génération basées sur Graviton2						
`db.t4g.medium`	2	4	8,13	Jusqu'à 2 085	Jusqu'à 5 Gbit/s	4.0.0 et 5.0.0
T3 — Classes d'instances de performance burstable de la génération précédente						
`db.t3.medium`	2	4	7,5	Jusqu'à 1 536	Jusqu'à 5 Gbit/s	3.6.0, 4.0.0 et 5.0.0

1. vCPU — Le nombre d'unités centrales virtuelles (CPU). Une UC virtuelle est une unité de capacité que vous pouvez utiliser pour comparer les classes d'instances. Au lieu d'acheter ou de louer un processeur particulier pour l'utiliser pendant plusieurs mois ou plusieurs années, vous louez la capacité à l'heure. Notre but est de fournir une quantité constante de capacité CPU sans que le matériel sous-jacent ait une influence.
2. Mémoire (GiB) : RAM, en gigaoctets, allouée à l'instance. Il existe souvent un ratio cohérent entre la mémoire et le processeur virtuel.
3. Stockage temporaire maximal (GiB) : RAM, en gigaoctets, allouée à l'instance pour le stockage de fichiers temporaires non persistants.
4. Bande passante maximale (Mbits/s) : bande passante maximale en mégabits par seconde. Divisez cette valeur par 8 pour calculer le débit attendu en mégaoctets par seconde.
5. Performances du réseau : vitesse du réseau par rapport aux autres classes d'instances.
6. Moteurs de support — Les moteurs Amazon DocumentDB qui prennent en charge la classe d'instance.

Identification du statut d'une instance

Pour connaître les statuts d'instance valides, leur signification et savoir comment identifier le statut de vos instances, consultez Surveillance de l'état d'une instance Amazon DocumentDB.

Cycle de vie des instances Amazon DocumentDB

Le cycle de vie d'une instance Amazon DocumentDB inclut la création, la modification, la maintenance et la mise à niveau, l'exécution de sauvegardes et de restaurations, le redémarrage et la suppression de l'instance. Cette section fournit des informations sur la façon de réaliser ces processus.

Rubriques

Vous pouvez créer une nouvelle instance Amazon DocumentDB à l'aide du AWS Management Console ou du. AWS CLI Pour ajouter une instance à un cluster, le cluster doit être dans un état available (disponible). Vous ne pouvez pas ajouter une instance à un cluster qui est arrêté. Si le cluster est arrêté, commencez par démarrer le cluster, attendez que le cluster devienne disponible, puis ajoutez une instance. Pour de plus amples informations, veuillez consulter Arrêt et démarrage d'un cluster Amazon DocumentDB.

> **Note**
>
> Si vous créez un cluster Amazon DocumentDB à l'aide de la console, une instance est automatiquement créée pour vous en même temps. Si vous souhaitez créer des instances supplémentaires, appliquez l'une des procédures suivantes.

Ajouter une instance Amazon DocumentDB à un cluster

Using the AWS Management Console

Utilisez la procédure suivante pour créer une instance pour votre cluster à l'aide de la console Amazon DocumentDB.

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.

2. Dans le panneau de navigation, choisissez Clusters.

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Pour choisir le cluster auquel vous voulez ajouter une instance, sélectionnez le bouton à gauche du nom du cluster.
4. Choisissez Actions, puis Add instances (Ajouter des instances).
5. Dans la page Add instance to: (Ajouter une instance à :)<cluster-name>, répétez les étapes suivantes pour chaque instance que vous souhaitez ajouter au cluster. Vous pouvez en avoir jusqu'à 15.

   a. Identifiant d'instance : vous pouvez saisir un identifiant unique pour cette instance ou autoriser Amazon DocumentDB à fournir l'identifiant d'instance en fonction de l'identifiant de cluster.

      Contraintes d'affectation de noms :

      - La longueur est de [1 à 63] lettres, chiffres ou traits d'union.
      - Le premier caractère doit être une lettre.
      - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
      - Doit être unique pour toutes les instances d'Amazon RDS, Neptune et Amazon Compte AWS DocumentDB par région.

   b. Classe d'instance : dans la liste déroulante, choisissez le type d'instance que vous souhaitez pour cette instance.

   c. Niveau de promotion : dans la liste déroulante, choisissez le niveau de promotion pour votre instance ou choisissez Aucune préférence pour permettre à Amazon DocumentDB de définir le niveau de promotion pour votre instance. Plus le chiffre est bas, plus la priorité est élevée. Pour de plus amples informations, veuillez consulter Contrôle de la cible du basculement.

   d. Pour ajouter d'autres instances, choisissez Add additional instances (Ajouter des instances supplémentaires) et répétez les étapes a, b et c.

6. Mettez fin à l'opération.

   - Pour ajouter des instances à votre cluster, choisissez Create (Créer).
   - Pour annuler l'opération, choisissez Annuler.

La création d'une instance prend quelques minutes. Vous pouvez utiliser la console ou AWS CLI consulter le statut de l'instance. Pour de plus amples informations, veuillez consulter Surveillance de l'état d'une instance.

Using the AWS CLI

Utilisez l'`create-db-instance` AWS CLI opération avec les paramètres suivants pour créer l'instance principale de votre cluster.

- **`--db-instance-class`** — Obligatoire. La capacité de calcul et de mémoire de l'instance, par exemple `db.m4.large`. Les classes d'instance ne sont pas toutes disponibles Régions AWS.
- **`--db-instance-identifier`** — Obligatoire. Chaîne en minuscules () qui identifie l'instance.

  Contraintes d'affectation de noms :

  - La longueur est de [1 à 63] lettres, chiffres ou traits d'union.
  - Le premier caractère doit être une lettre.
  - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
  - Doit être unique pour toutes les instances d'Amazon RDS, Neptune et Amazon Compte AWS DocumentDB par région.
- **`--engine`** — Obligatoire. Doit indiquer `docdb`.
- **`--availability-zone`**— Facultatif. La zone de disponibilité dans laquelle vous voulez créer l'instance. Utilisez ce paramètre pour localiser vos instances dans différentes zones de disponibilité, afin d'accroître la tolérance aux pannes. Pour de plus amples informations, veuillez consulter Haute disponibilité et réplication Amazon DocumentDB.
- **`--promotion-tier`**— Facultatif. Le niveau de priorité de basculement pour cette instance. Il doit être compris entre 0 et 15, les chiffres les plus bas indiquant une priorité plus élevée. Pour de plus amples informations, veuillez consulter Contrôle de la cible du basculement.

1. Tout d'abord, déterminez dans quelles zones de disponibilité vous pouvez créer votre instance.

Si vous souhaitez spécifier la zone de disponibilité avant de créer votre instance, exécutez la commande suivante pour déterminer quelles zones de disponibilité sont disponibles pour votre cluster Amazon DocumentDB.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-clusters \
       --query 'DBClusters[*].[DBClusterIdentifier,AvailabilityZones[*]]'
```

Pour Windows :

```
aws docdb describe-db-clusters ^
       --query 'DBClusters[*].[DBClusterIdentifier,AvailabilityZones[*]]'
```

Le résultat de cette opération ressemble à ceci.

```
[
    [
        "sample-cluster",
        [
            "us-east-1c",
            "us-east-1b",
            "us-east-1a"
        ]
    ]
]
```

2. Ensuite, identifiez les classes d'instance que vous pouvez créer dans votre région.

   Pour déterminer quelles classes d'instances sont disponibles dans votre région, exécutez la commande suivante. Dans la sortie, choisissez une classe d'instance pour l'instance que vous souhaitez ajouter à votre cluster Amazon DocumentDB.

   Pour Linux, macOS ou Unix :

```
aws docdb describe-orderable-db-instance-options \
        --engine docdb \
        --query 'OrderableDBInstanceOptions[*].DBInstanceClass'
```

Pour Windows :

```
aws docdb describe-orderable-db-instance-options ^
        --engine docdb ^
        --query 'OrderableDBInstanceOptions[*].DBInstanceClass'
```

Le résultat de cette opération ressemble à ceci.

```
[
    "db.r5.16xlarge",
    "db.r5.2xlarge",
    "db.r5.4xlarge",
    "db.r5.8xlarge",
    "db.r5.large",
    "db.r5.xlarge"
]
```

3. Enfin, ajoutez une instance à votre cluster Amazon DocumentDB.

   Pour ajouter une instance à votre cluster Amazon DocumentDB, exécutez la commande suivante.

   Pour Linux, macOS ou Unix :

```
aws docdb create-db-instance \
       --db-cluster-identifier sample-cluster \
       --db-instance-identifier sample-instance-2 \
       --availability-zone us-east-1b \
       --promotion-tier 2 \
       --db-instance-class db.r5.xlarge \
       --engine docdb
```

   Pour Windows :

```
aws docdb create-db-instance ^
       --db-cluster-identifier sample-cluster ^
       --db-instance-identifier sample-instance-2 ^
       --availability-zone us-east-1b ^
       --promotion-tier 2 ^
       --db-instance-class db.r5.xlarge ^
```

```
        --engine docdb
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "sample-instance-2",
        "DBInstanceClass": "db.r5.xlarge",
        "Engine": "docdb",
        "DBInstanceStatus": "creating",
        "PreferredBackupWindow": "02:00-02:30",
        "BackupRetentionPeriod": 1,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
            }
        ],
        "AvailabilityZone": "us-east-1b",
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-6242c31a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-abcd0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-wxyz0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sun:11:35-sun:12:05",
        "PendingModifiedValues": {},
```

```
        "EngineVersion": "3.6.0",
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "DBClusterIdentifier": "sample-cluster",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "CACertificateIdentifier": "rds-ca-2019",
        "PromotionTier": 2,
        "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-instance-2"
    }
}
```

La création de l'instance prend quelques minutes. Vous pouvez utiliser la console ou AWS CLI consulter le statut de l'instance. Pour de plus amples informations, veuillez consulter Surveillance de l'état d'une instance Amazon DocumentDB.

Décrire les instances Amazon DocumentDB

Vous pouvez utiliser la console de gestion Amazon DocumentDB ou le AWS CLI pour consulter des informations telles que les points de terminaison de connexion, les groupes de sécurité (VPC), l'autorité de certification et les groupes de paramètres relatifs à vos instances Amazon DocumentDB.

Using the AWS Management Console

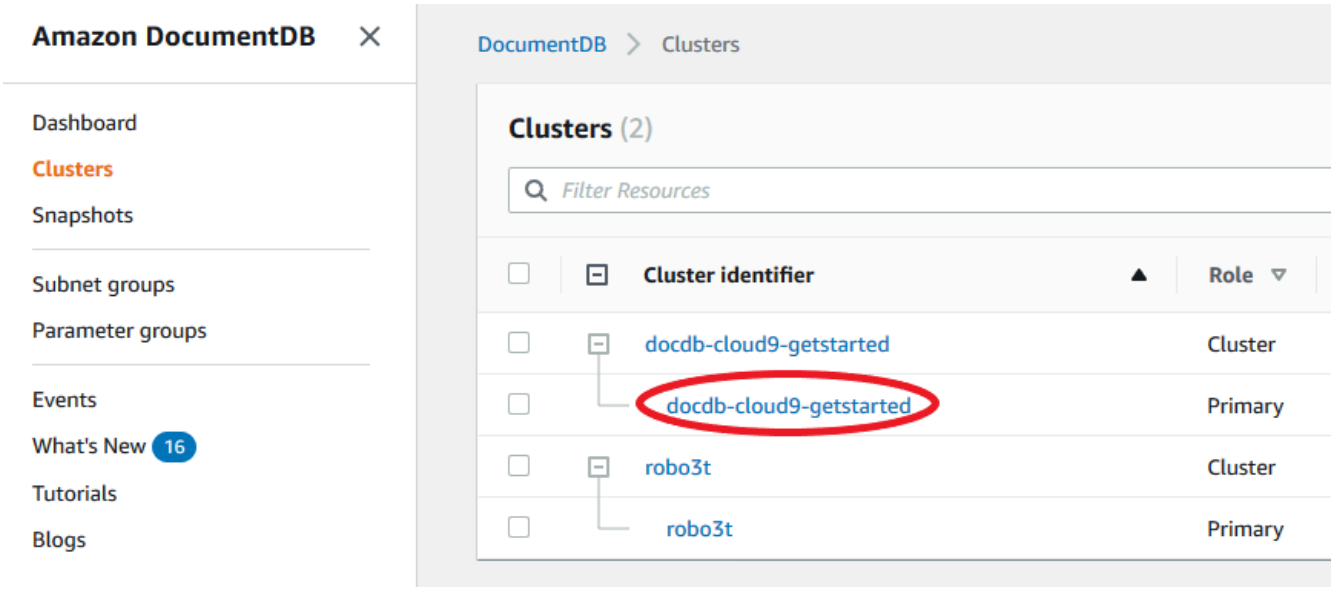
Pour afficher les détails de vos instances à l'aide de l’ AWS Management Console, suivez les étapes ci-dessous.

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu

(≡)
dans le coin supérieur gauche de la page.

3. Dans la boîte de navigation Clusters, vous verrez la colonne Cluster Identifier. Vos instances sont répertoriées sous des clusters, comme dans la capture d'écran ci-dessous.



4. Dans la liste des instances, choisissez le nom de l'instance dont vous souhaitez voir les détails. Les informations relatives à l'instance sont organisées dans les groupes suivants :

   - Résumé : informations générales sur l'instance, notamment la version du moteur, la classe, le statut et toute maintenance en attente.
   - Connectivité et sécurité : la section Connect répertorie les points de terminaison de connexion permettant de se connecter à cette instance avec le shell mongo ou une application. La section Security Groups (Groupes de sécurité) répertorie les groupes de sécurité associés à cette instance ainsi que leur ID et de VPC et leurs descriptions.
   - Configuration : la section Détails répertorie les configurations et le statut de l'instance, y compris le nom de ressource Amazon (ARN), le point de terminaison, le rôle, la classe et l'autorité de certification de l'instance. Elle répertorie également les paramètres de sécurité et de réseau de l'instance, ainsi que les informations de sauvegarde. La section Cluster details (Détails du cluster) répertorie les détails du cluster auquel cette instance appartient. La section Cluster instances (Instances de cluster) répertorie toutes les instances appartenant à votre cluster avec l'état du rôle et du groupe de paramètres de cluster de chaque instance.

> **Note**
>
> Vous pouvez modifier le cluster associé à votre instance en sélectionnant Modify (Modifier) en regard de l'en-tête Cluster details (Détails du cluster). Pour de plus amples informations, veuillez consulter Modification d'un cluster Amazon DocumentDB.

- Surveillance : CloudWatch enregistre les métriques pour cette instance. Pour de plus amples informations, veuillez consulter Surveillance d'Amazon DocumentDB avec CloudWatch.
- Événements et tags : la section Événements récents répertorie les événements récents pour cette instance. Amazon DocumentDB conserve un enregistrement des événements liés à vos clusters, instances, instantanés, groupes de sécurité et groupes de paramètres de cluster. Ces informations comprennent la date, l'heure et le message associés à chaque événement. La section Tags (Balises) répertorie les balises attachées à ce cluster. Pour de plus amples informations, veuillez consulter Balisage des ressources Amazon DocumentDB.

Using the AWS CLI

Pour consulter les détails de vos instances Amazon DocumentDB à l'aide de AWS CLI, utilisez la `describe-db-clusters` commande comme indiqué dans les exemples ci-dessous. Pour plus d'informations, consultez le document `DescribeDBInstances`de référence de l'API de gestion des ressources Amazon DocumentDB.

> **Note**
>
> Pour certaines fonctionnalités de gestion telles que la gestion du cycle de vie des clusters et des instances, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon RDS. Le paramètre de `filterName=engine,Values=docdb` filtre renvoie uniquement les clusters Amazon DocumentDB.

1. Répertoriez toutes les instances Amazon DocumentDB.

   Le AWS CLI code suivant répertorie les détails de toutes les instances Amazon DocumentDB d'une région.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-instances \
    --filter Name=engine,Values=docdb
```

Pour Windows :

```
aws docdb describe-db-instances \
    --filter Name=engine,Values=docdb
```

2. Répertorier tous les détails d'une instance Amazon DocumentDB spécifiée

Le code suivant répertorie les détails de `sample-cluster-instance`. L'inclusion du paramètre `--db-instance-identifier` avec le nom d'une instance limite la sortie aux informations sur cette instance particulière.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-instances \
    --db-instance-identifier sample-cluster-instance
```

Pour Windows :

```
aws docdb describe-db-instances \
    --db-instance-identifier sample-cluster-instance
```

Le résultat de cette opération ressemble à ce qui suit.

```
{
    "DBInstances": [
        {
            "DbiResourceId": "db-BJKKB54PIDV5QFKGVRX5T3S6GM",
            "DBInstanceArn": "arn:aws:rds:us-east-1:012345678901:db:sample-
cluster-instance-00",
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-77186e0d",
                    "Status": "active"
                }
            ],
```

```
            "DBInstanceClass": "db.r5.large",
            "DBInstanceStatus": "creating",
            "AutoMinorVersionUpgrade": true,
            "PreferredMaintenanceWindow": "fri:09:32-fri:10:02",
            "BackupRetentionPeriod": 1,
            "StorageEncrypted": true,
            "DBClusterIdentifier": "sample-cluster",
            "EngineVersion": "3.6.0",
            "AvailabilityZone": "us-east-1a",
            "Engine": "docdb",
            "PromotionTier": 2,
            "DBInstanceIdentifier": "sample-cluster-instance",
            "PreferredBackupWindow": "00:00-00:30",
            "PubliclyAccessible": false,
            "DBSubnetGroup": {
                "DBSubnetGroupName": "default",
                "Subnets": [
                    {
                        "SubnetIdentifier": "subnet-4e26d263",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1a"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-afc329f4",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1c"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-b3806e8f",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1e"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-53ab3636",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1d"
                        },
                        "SubnetStatus": "Active"
```

```
                    },
                    {
                        "SubnetIdentifier": "subnet-991cb8d0",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1b"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-29ab1025",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1f"
                        },
                        "SubnetStatus": "Active"
                    }
                ],
                "VpcId": "vpc-91280df6",
                "DBSubnetGroupDescription": "default",
                "SubnetGroupStatus": "Complete"
            },
            "PendingModifiedValues": {},
            "KmsKeyId": "arn:aws:kms:us-east-1:012345678901:key/0961325d-
a50b-44d4-b6a0-a177d5ff730b"
        }
    ]
}
```

Modification d'une instance Amazon DocumentDB

Vous pouvez modifier votre instance Amazon DocumentDB à l'aide du AWS Management Console ou du. AWS CLI Pour modifier une instance, l'instance doit être dans l'état available (disponible). Vous ne pouvez pas modifier une instance qui est arrêtée. Si le cluster est arrêté, commencez par démarrer le cluster, attendez que l'instance devienne disponible, puis apportez les modifications souhaitées. Pour de plus amples informations, veuillez consulter Arrêt et démarrage d'un cluster Amazon DocumentDB.

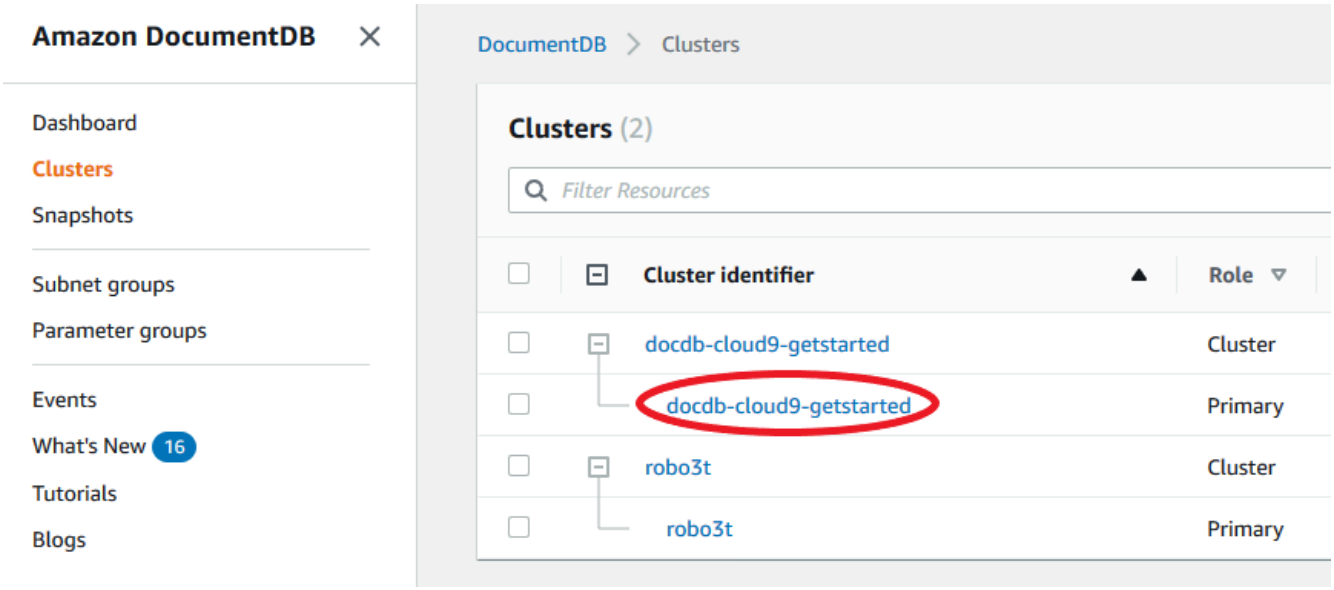
Using the AWS Management Console

Pour modifier une instance Amazon DocumentDB spécifique à l'aide de la console, procédez comme suit.

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans la boîte de navigation Clusters, vous verrez la colonne Cluster Identifier. Vos instances sont répertoriées sous des clusters, comme dans la capture d'écran ci-dessous.



4. Cochez la case située à gauche de l'instance que vous souhaitez modifier.
5. Choisissez Actions, puis Modify (Modifier).
6. Dans le volet Modify instance: <instance-name> (Modifier l'instance : <nom-instance>), apportez les modifications souhaitées. Vous pouvez apporter les modifications suivantes :

   - Spécifications de l'instance : identifiant et classe de l'instance. Contraintes d'attribution de noms aux identifiants d'instance :
     - Identifiant d'instance — Entrez un nom unique pour toutes les instances que vous possédez Compte AWS dans la région actuelle. L'identifiant d'instance doit contenir [1 à 63] caractères alphanumériques ou traits d'union, avoir une lettre comme premier caractère et ne peut pas se terminer par un tiret ni contenir deux tirets consécutifs.

- Classe d'instance : dans le menu déroulant, sélectionnez une classe d'instance pour votre instance Amazon DocumentDB. Pour de plus amples informations, veuillez consulter Gestion de classes d'instance.

- Autorité de certification : certificat de serveur pour cette instance. Pour de plus amples informations, veuillez consulter Mise à jour de vos certificats TLS Amazon DocumentDB.
- Basculement : lors du basculement, l'instance présentant le niveau de promotion le plus élevé sera promue au niveau principal. Pour de plus amples informations, veuillez consulter Basculement Amazon DocumentDB.
- Maintenance : fenêtre de maintenance au cours de laquelle les modifications ou correctifs en attente sont appliqués aux instances du cluster.

7. Lorsque vous avez terminé, choisissez Continue (continuer) pour afficher un récapitulatif de vos modifications.
8. Après vérification de vos modifications, vous pouvez les appliquer immédiatement ou au cours de la fenêtre de maintenance suivante sous Scheduling of modifications (Planification des modifications). Choisissez Modify instance (Modifier l'instance) pour enregistrer vos modifications. Vous pouvez également choisir Cancel (Annuler) pour ignorer vos modifications.

L'application de vos modifications prend quelques minutes. Vous pouvez uniquement utiliser l'instance lorsqu'elle présente le statut disponible. Vous pouvez surveiller l'état de l'instance en utilisant la console ou la AWS CLI. Pour de plus amples informations, veuillez consulter Surveillance de l'état d'une instance Amazon DocumentDB.

Using the AWS CLI

Pour modifier une instance Amazon DocumentDB spécifique à l'aide du AWS CLI, utilisez le `modify-db-instance` avec les paramètres suivants. Pour plus d'informations, consultez ModifyDBInstance. Le code suivant modifie la classe d'instance en `db.r5.large` pour l'instance `sample-instance`.

Paramètres

- **`--db-instance-identifier`** — Obligatoire. Identifiant de l'instance à modifier.
- **`--db-instance-class`**— Facultatif. La nouvelle capacité de calcul et de mémoire de l'instance ; par exemple,`db.r5.large`. Les classes d'instance ne sont pas toutes disponibles Régions AWS. Si vous modifiez la classe d'instance, une panne se produit lors de la

modification. La modification est appliquée lors de la fenêtre de maintenance suivante, sauf si `ApplyImmediately` elle est spécifiée comme vraie pour cette demande.

- **`--apply-immediately`**ou **`--no-apply-immediately`** — Facultatif. Indique si cette modification doit être appliquée immédiatement ou s'il faut attendre le prochain créneau de maintenance. Si ce paramètre n'est pas appliqué, la modification est réalisée au cours du créneau de maintenance suivant.

Example

Pour Linux, macOS ou Unix :

```
aws docdb modify-db-instance \
       --db-instance-identifier sample-instance \
       --db-instance-class db.r5.large \
       --apply-immediately
```

Pour Windows :

```
aws docdb modify-db-instance ^
       --db-instance-identifier sample-instance ^
       --db-instance-class db.r5.large ^
       --apply-immediately
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBInstances": [
        {
            "DBInstanceIdentifier": "sample-instance-1",
            "DBInstanceClass": "db.r5.large",
            "Engine": "docdb",
            "DBInstanceStatus": "modifying",
            "Endpoint": {
                "Address": "sample-instance-1.node.us-east-1.docdb.amazonaws.com",
                "Port": 27017,
                "HostedZoneId": "ABCDEFGHIJKLM"
            },
            "InstanceCreateTime": "2020-01-10T22:18:55.921Z",
            "PreferredBackupWindow": "02:00-02:30",
            "BackupRetentionPeriod": 1,
            "VpcSecurityGroups": [
```

```
                {
                    "VpcSecurityGroupId": "sg-abcd0123",
                    "Status": "active"
                }
            ],
            "AvailabilityZone": "us-east-1a",
            "DBSubnetGroup": {
                "DBSubnetGroupName": "default",
                "DBSubnetGroupDescription": "default",
                "VpcId": "vpc-abcd0123",
                "SubnetGroupStatus": "Complete",
                "Subnets": [
                    {
                        "SubnetIdentifier": "subnet-abcd0123",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1a"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-abcd0123",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1b"
                        },
                        "SubnetStatus": "Active"
                    }
                ]
            },
            "PreferredMaintenanceWindow": "sun:10:57-sun:11:27",
            "PendingModifiedValues": {
                "DBInstanceClass": "db.r5.large"
            },
            "EngineVersion": "3.6.0",
            "AutoMinorVersionUpgrade": true,
            "PubliclyAccessible": false,
            "DBClusterIdentifier": "sample-cluster",
            "StorageEncrypted": true,
            "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/wJalrXUtnFEMI/
K7MDENG/bPxRfiCYEXAMPLEKEY",
            "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
            "CACertificateIdentifier": "rds-ca-2019",
            "PromotionTier": 1,
            "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:sample-
instance-1",
```

```
            "EnabledCloudwatchLogsExports": [
                "profiler"
            ]
        }
    ]
}
```

L'application de vos modifications prend quelques minutes. Vous pouvez uniquement utiliser l'instance lorsqu'elle présente le statut disponible. Vous pouvez surveiller l'état de l'instance à l'aide du AWS Management Console ou AWS CLI. Pour de plus amples informations, veuillez consulter Surveillance de l'état d'une instance Amazon DocumentDB.

Redémarrage d'une instance Amazon DocumentDB

Il se peut que vous deviez redémarrer votre instance Amazon DocumentDB, généralement pour des raisons de maintenance. Par exemple, si vous effectuez certaines modifications, telles que la modification du groupe de paramètres associé à un cluster, vous devez redémarrer les instances du cluster pour que ces modifications prennent effet. Vous pouvez redémarrer une instance spécifiée à l'aide du AWS Management Console ou du AWS CLI.

Le redémarrage d'une instance entraîne celui du service du moteur de base de données. Le redémarrage d'une instance entraîne une interruption momentanée, au cours de laquelle le statut de l'instance est défini sur `rebooting`. Un événement Amazon DocumentDB est créé lorsque le redémarrage est terminé.

Le redémarrage d'une instance n'entraîne pas de basculement. Pour basculer un cluster Amazon DocumentDB, utilisez AWS Management Console l'opération ou AWS CLI . `failover-db-cluster` Pour de plus amples informations, veuillez consulter Basculement Amazon DocumentDB.

Vous ne pouvez pas redémarre votre instance si celle-ci n'est pas à l'état disponible. Votre base de données peut être indisponible pour plusieurs raisons, notamment pour cause de modification demandée précédemment ou d'action de créneau de maintenance. Pour de plus amples informations sur les états d'instance, veuillez consulter Surveillance de l'état d'une instance Amazon DocumentDB.

Using the AWS Management Console

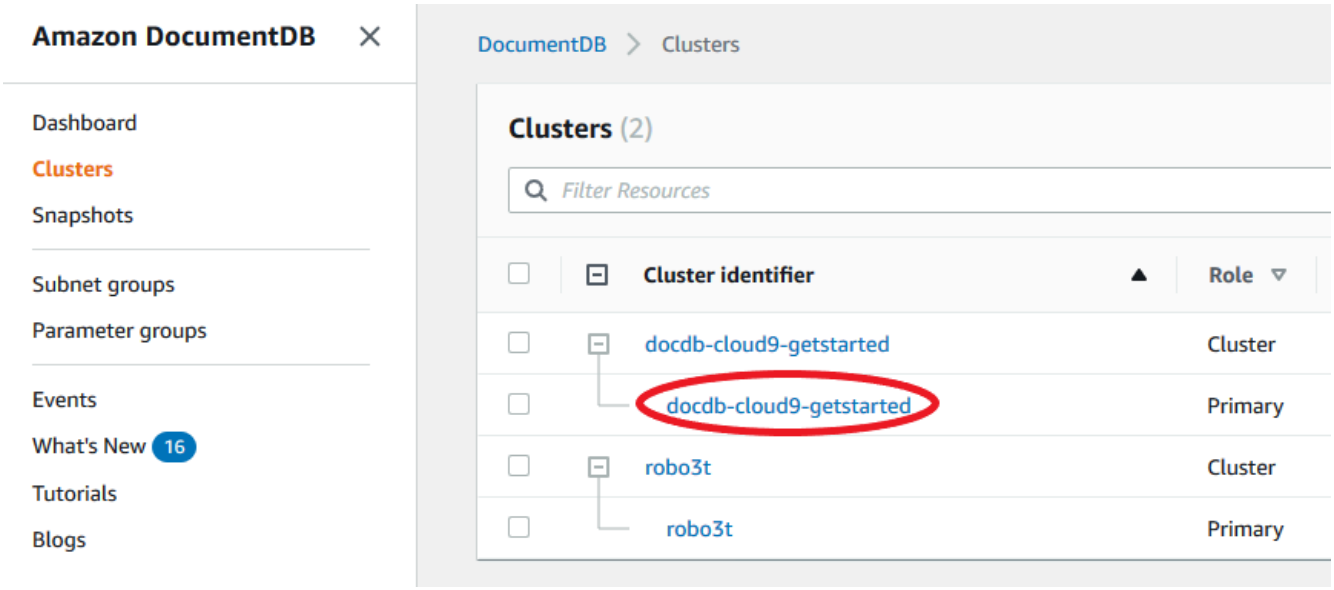
La procédure suivante redémarre l'instance que vous spécifiez à l'aide de la console.

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.

2. Dans le panneau de navigation, choisissez Clusters.

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans la boîte de navigation Clusters, vous verrez la colonne Cluster Identifier. Vos instances sont répertoriées sous des clusters, comme dans la capture d'écran ci-dessous.



4. Cochez la case située à gauche de l'instance que vous souhaitez redémarrer.
5. Choisissez Actions, Reboot (Redémarrer), puis Reboot (Redémarrer) pour confirmer votre redémarrage.

Le redémarrage de votre instance prend quelques minutes. Vous pouvez uniquement utiliser l'instance lorsqu'elle présente le statut disponible. Vous pouvez surveiller le statut de l'instance en utilisant la console ou la AWS CLI. Pour de plus amples informations, veuillez consulter Surveillance de l'état d'une instance Amazon DocumentDB.

Using the AWS CLI

Pour redémarrer une instance Amazon DocumentDB, utilisez l'`reboot-db-instance`opération avec le `--db-instance-identifier` paramètre. Ce paramètre spécifie l'identifiant de l'instance à redémarrer.

Le code suivant redémarre l'instance `sample-instance`.

Example

Pour Linux, macOS ou Unix :

```
aws docdb reboot-db-instance \
       --db-instance-identifier sample-instance
```

Pour Windows :

```
aws docdb reboot-db-instance ^
       --db-instance-identifier sample-instance
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "sample-instance",
        "DBInstanceClass": "db.r5.large",
        "Engine": "docdb",
        "DBInstanceStatus": "rebooting",
        "Endpoint": {
            "Address": "sample-instance.node.us-east-1.docdb.amazonaws.com",
            "Port": 27017,
            "HostedZoneId": "ABCDEFGHIJKLM"
        },
        "InstanceCreateTime": "2020-03-27T08:05:56.314Z",
        "PreferredBackupWindow": "02:00-02:30",
        "BackupRetentionPeriod": 1,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
            }
        ],
        "AvailabilityZone": "us-east-1c",
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-abcd0123",
            "SubnetGroupStatus": "Complete",
```

```
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-abcd0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-wxyz0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sun:06:53-sun:07:23",
        "PendingModifiedValues": {},
        "EngineVersion": "3.6.0",
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "DBClusterIdentifier": "sample-cluster",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "CACertificateIdentifier": "rds-ca-2019",
        "PromotionTier": 1,
        "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-instance",
        "EnabledCloudwatchLogsExports": [
            "profiler"
        ]
    }
}
```

Le redémarrage de votre instance prend quelques minutes. Vous pouvez uniquement utiliser l'instance lorsqu'elle présente le statut disponible. Vous pouvez surveiller l'état de l'instance en utilisant la console ou la AWS CLI. Pour de plus amples informations, veuillez consulter Surveillance de l'état d'une instance Amazon DocumentDB.

Supprimer une instance Amazon DocumentDB

Vous pouvez supprimer votre instance Amazon DocumentDB à l'aide du AWS Management Console ou du. AWS CLI Pour supprimer une instance, l'instance doit être à l'état available (disponible). Vous ne pouvez pas supprimer une instance qui est arrêtée. Si le cluster Amazon DocumentDB qui contient votre instance est arrêté, démarrez d'abord le cluster, attendez qu'elle soit disponible, puis supprimez-la. Pour de plus amples informations, veuillez consulter Arrêt et démarrage d'un cluster Amazon DocumentDB.

> **Note**
>
> Amazon DocumentDB stocke toutes vos données dans le volume du cluster. Les données restent dans ce volume de cluster, même si vous supprimez toutes les instances de votre cluster. Si vous avez besoin d'accéder aux données à nouveau, vous pouvez ajouter une instance au cluster à tout moment et reprendre votre activité là où vous l'aviez laissée.

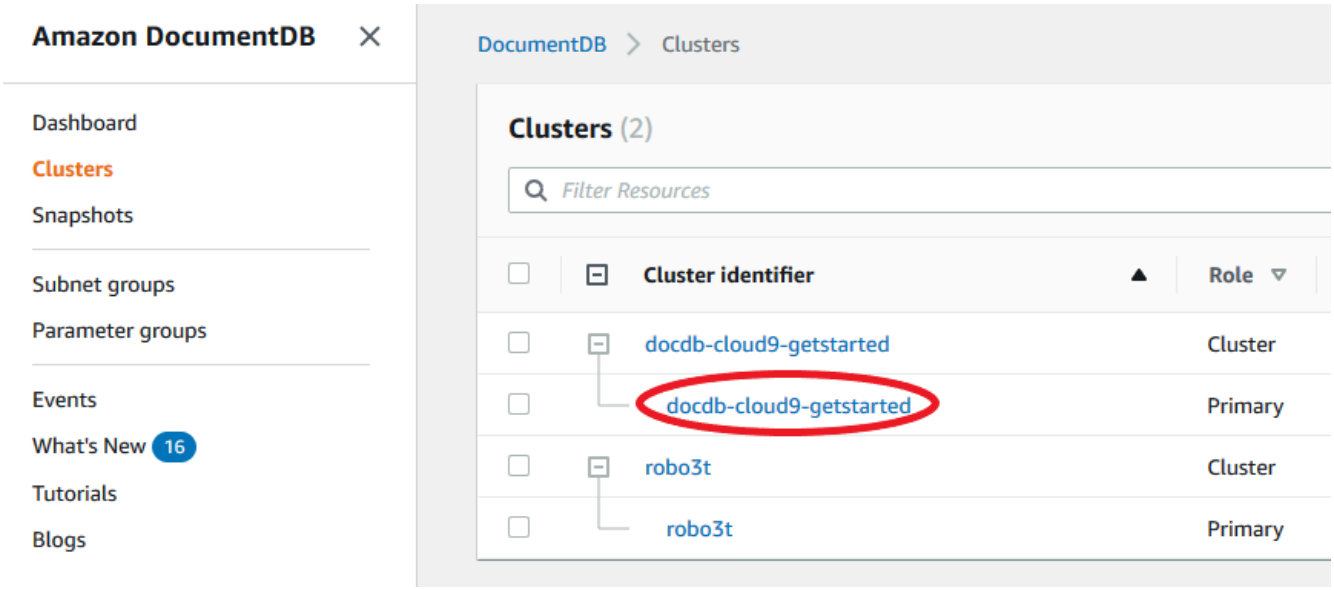
Using the AWS Management Console

La procédure suivante supprime une instance Amazon DocumentDB spécifiée à l'aide de la console.

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans la boîte de navigation Clusters, vous verrez la colonne Cluster Identifier. Vos instances sont répertoriées sous des clusters, comme dans la capture d'écran ci-dessous.



4. Cochez la case située à gauche de l'instance que vous souhaitez supprimer.
5. Choisissez Actions, puis Delete (Supprimer).

    1. Si vous supprimez la dernière instance de votre cluster :

        - Create final cluster snapshot? (Créer un instantané final de cluster ?) — Choisissez Oui si vous souhaitez créer un instantané final avant que le cluster ne soit supprimé. Sinon, choisissez Non.
        - Nom du cliché final : si vous choisissez de créer un instantané final, entrez l'identifiant du cliché de cluster du nouveau cliché de cluster créé.
        - Delete <instance-name> instance? (Supprimer l'instance <nom-instance> ?) — Entrez la phrase supprimer le cluster entier dans le champ pour confirmer la suppression.

    2. Si vous ne supprimez pas la dernière instance de votre cluster :

        - Delete <instance-name> instance? (Supprimer l'instance <nom-instance> ?) — Entrez la phrase supprimez-moi dans le champ pour confirmer la suppression.
6. Sélectionnez Delete (Supprimer) pour supprimer l'instance.

La suppression de l'instance prend plusieurs minutes. Pour surveiller l'état d'une instance, consultez Surveillance de l'état d'une instance Amazon DocumentDB.

Using the AWS CLI

La procédure suivante supprime une instance Amazon DocumentDB à l'aide du. AWS CLI

1. Tout d'abord, déterminez le nombre d'instances présentes dans votre cluster Amazon DocumentDB :

   Pour déterminer le nombre d'instances de votre cluster, exécutez la commande `describe-db-clusters`, comme suit.

   ```
   aws docdb describe-db-clusters \
       --db-cluster-identifier sample-cluster \
       --query 'DBClusters[*].
   [DBClusterIdentifier,DBClusterMembers[*].DBInstanceIdentifier]'
   ```

   Le résultat de cette opération ressemble à ceci.

   ```
   [
       [
           "sample-cluster",
           [
               "sample-instance-1",
               "sample-instance-2"
           ]
       ]
   ]
   ```

2. S'il existe plusieurs instances dans votre cluster Amazon DocumentDB :

   Pour supprimer une instance Amazon DocumentDB spécifiée, utilisez la `delete-db-instance` commande avec le `--db-instance-identifier` paramètre, comme indiqué ci-dessous. La suppression de l'instance prend plusieurs minutes. Pour surveiller l'état d'une instance, consultez Surveillance de l'état d'une instance Amazon DocumentDB.

   ```
   aws docdb delete-db-instance \
          --db-instance-identifier sample-instance-2
   ```

   Le résultat de cette opération ressemble à ceci.

   ```
   {
       "DBInstance": {
           "DBInstanceIdentifier": "sample-instance-2",
           "DBInstanceClass": "db.r5.large",
           "Engine": "docdb",
           "DBInstanceStatus": "deleting",
   ```

```
        "Endpoint": {
            "Address": "sample-instance-2.node.us-east-1.docdb.amazonaws.com",
            "Port": 27017,
            "HostedZoneId": "ABCDEFGHIJKLM"
        },
        "InstanceCreateTime": "2020-03-27T08:05:56.314Z",
        "PreferredBackupWindow": "02:00-02:30",
        "BackupRetentionPeriod": 1,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
            }
        ],
        "AvailabilityZone": "us-east-1c",
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-6242c31a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-abcd0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-wxyz0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sun:06:53-sun:07:23",
        "PendingModifiedValues": {},
        "EngineVersion": "3.6.0",
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "DBClusterIdentifier": "sample-cluster",
        "StorageEncrypted": true,
```

```
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "CACertificateIdentifier": "rds-ca-2019",
        "PromotionTier": 1,
        "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-instance-2",
        "EnabledCloudwatchLogsExports": [
            "profiler"
        ]
    }
}
```

3. Si l'instance que vous souhaitez supprimer est la dernière instance de votre cluster Amazon DocumentDB :

   Si vous supprimez la dernière instance d'un cluster Amazon DocumentDB, vous supprimez également ce cluster ainsi que les instantanés automatiques et les sauvegardes continues associés à ce cluster.

   Pour supprimer la dernière instance de votre cluster, vous pouvez supprimer le cluster et éventuellement créer un instantané final. Pour de plus amples informations, veuillez consulter Suppression d'un cluster Amazon DocumentDB.

Deletion protection (Protection contre la suppression)

La suppression de la dernière instance d'un cluster Amazon DocumentDB entraîne également la suppression du cluster, ainsi que des instantanés automatiques et des sauvegardes continues associés à ce cluster. Amazon DocumentDB applique la protection contre la suppression à un cluster, que vous exécutiez l'opération de suppression à l'aide du ou du AWS Management Console . AWS CLI Si la protection contre la suppression est activée, vous ne pouvez pas supprimer de cluster.

Pour supprimer un cluster pour lequel la protection contre la suppression est activée, vous devez commencer par modifier le cluster et désactiver la protection contre la suppression. Pour de plus amples informations, veuillez consulter Suppression d'un cluster Amazon DocumentDB.

Gestion des groupes de sous-réseaux Amazon DocumentDB

Un cloud privé virtuel (VPC) est un réseau virtuel dédié à votreCompte AWS. Il est logiquement isolé des autres réseaux virtuels dans le cloud AWS. Vous pouvez lancer vosAWS ressources, telles que des clusters Amazon DocumentDB, dans votre Amazon VPC. Vous pouvez spécifier une plage

d'adresses IP pour le VPC, ajouter des sous-réseaux, associer des groupes de sécurité et configurer des tables de routage.

Un sous-réseau est une plage d'adresses IP dans votre Amazon VPC. Vous pouvez lancer des ressources AWS dans un sous-réseau spécifié. Utilisez un sous-réseau public pour les ressources qui doivent être connectées à Internet. Utilisez un sous-réseau privé pour les ressources qui ne doivent pas être connectées à Internet. Pour de plus amples informations sur les sous-réseaux publics et privés, veuillez consulter Principes de base des VPC et des sous-réseaux dans le Guide de l'utilisateur Amazon Virtual Private Cloud.

Un groupe de sous-réseaux DB est une collection de sous-réseaux que vous créez dans un VPC et que vous spécifiez alors pour vos clusters. Un groupe de sous-réseaux vous permet de spécifier un VPC particulier lors de la création de clusters. Si vous utilisez le groupe de sous-réseaux `default`, ce dernier couvre tous les sous-réseaux dans le VPC.

Chaque groupe de sous-réseaux DB doit avoir des sous-réseaux dans au moins deux zones de disponibilité d'une région donnée. Lors de la création d'un cluster de base de données dans un VPC, vous devez sélectionner un groupe de sous-réseaux DB. Amazon DocumentDB utilise ce groupe de sous-réseaux DB et votre zone de disponibilité privilégiée, pour sélectionner dans ce sous-réseau un sous-réseau et une adresse IP à associer à votre cluster. Si l'instance principale échoue, Amazon DocumentDB peut promouvoir une instance de réplique correspondante en tant que nouvelle instance principale. Le service peut ensuite créer une nouvelle instance de réplica à l'aide d'une adresse IP du sous-réseau dans lequel l'ancienne instance principale était située.

Quand Amazon DocumentDB crée une instance dans un VPC, il affecte une interface réseau à votre cluster en utilisant une adresse IP de votre groupe de sous-réseaux de base de données. Nous vous recommandons fortement d'utiliser le nom DNS car l'adresse IP sous-jacente peut changer pendant le basculement. Pour plus d'informations, veuillez consulter Points de terminaison Amazon DocumentDB.

Pour plus d'informations sur la création de votre propre VPC et de vos propres sous-réseaux, consultez la section Utilisation des VPC et des sous-réseaux dans le guide de l'utilisateur d'Amazon Virtual Private Cloud.

Rubriques

- Création d'un groupe de sous-réseaux Amazon DocumentDB
- Description d'un groupe de sous-réseaux Amazon DocumentDB
- Modification d'un groupe de sous-réseaux Amazon DocumentDB

- Suppression d'un groupe de sous-réseaux Amazon DocumentDB

Création d'un groupe de sous-réseaux Amazon DocumentDB

Lorsque vous créez un cluster Amazon DocumentDB, vous devez choisir un Amazon VPC et le groupe de sous-réseaux correspondant au sein de cet Amazon VPC pour lancer votre cluster. Les sous-réseaux déterminent la zone de disponibilité et la plage d'adresses IP au sein de la zone de disponibilité que vous souhaitez utiliser pour lancer une instance.

Un groupe de sous-réseaux est un ensemble nommé de sous-réseaux (ou AZ) qui vous permet de spécifier les zones de disponibilité que vous souhaitez utiliser pour lancer des instances Amazon DocumentDB. Par exemple, dans un cluster comportant trois instances, il est recommandé de provisionner chacune de ces instances dans des AZ distincts, afin d'optimiser la haute disponibilité. Ainsi, si une seule AZ échoue, cela n'affectera qu'une seule instance.

Actuellement, les instances Amazon DocumentDB peuvent être mises en service dans un maximum de trois AZ. Même si un groupe de sous-réseaux comporte plus de trois sous-réseaux, vous ne pourrez utiliser que trois de ces sous-réseaux pour créer un cluster Amazon DocumentDB. Par conséquent, lorsque vous créez un groupe de sous-réseaux, nous vous recommandons de ne choisir que les trois sous-réseaux dont vous souhaitez déployer vos instances.

Par exemple : un cluster est créé et Amazon DocumentDB choisit AZs {1A, 1B et 1C}. Si vous essayez de créer une instance dans la zone de disponibilité {1D}, l'appel d'API échoue. Toutefois, si vous choisissez de créer une instance, sans spécifier la zone de référence particulière, Amazon DocumentDB choisira une instance en votre nom. Amazon DocumentDB utilise un algorithme pour équilibrer la charge des instances entre les AZ afin de vous aider à atteindre une haute disponibilité. Si trois instances sont provisionnées, par défaut, elles seront provisionnées sur trois AZ et ne seront pas toutes provisionnées dans une seule AZ.

Bonnes pratiques

- Sauf si vous avez une raison précise, créez toujours un groupe de sous-réseaux avec trois sous-réseaux. Cela garantit que les clusters comportant trois instances ou plus seront en mesure d'atteindre une disponibilité accrue, car les instances seront provisionnées sur trois AZ.
- Répartissez toujours les instances sur plusieurs zones de disponibilité pour obtenir une haute disponibilité. Ne placez jamais toutes les instances d'un cluster dans une seule zone de disponibilité.

- Des événements de basculement pouvant se produire à n'importe quel moment, vous ne devez pas supposer qu'une instance principale ou des instances de réplica seront toujours placées dans une zone de disponibilité spécifique.

Comment créer un groupe de sous-réseaux

Vous pouvez utiliser leAWS Management Console ouAWS CLI pour créer un groupe de sous-réseaux Amazon DocumentDB :

Using the AWS Management Console

Pour créer un groupe de sous-réseaux Amazon DocumentDB.

Pour créer un groupe de sous-réseaux Amazon DocumentDB

1. Connectez-vous à laAWS Management Console et ouvrez la console Amazon DocumentDB à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Groupes de sous-réseaux, puis Créer.

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Sur la page Create subnet group (Créer un groupe de sous-réseaux) :
   a. Dans la section Subnet group details (Informations sur le groupe de sous-réseaux) :
      i. Nom : entrez un nom significatif pour le groupe de sous-réseaux.
      ii. Description – Saisissez une description pour le groupe de sous-réseaux.
   b. Dans la section Add subnets (Ajouter des sous-réseaux) :
      i. VPC —Dans la liste, choisissez un VPC pour ce groupe de sous-réseaux.
      ii. Effectuez l'une des actions suivantes :

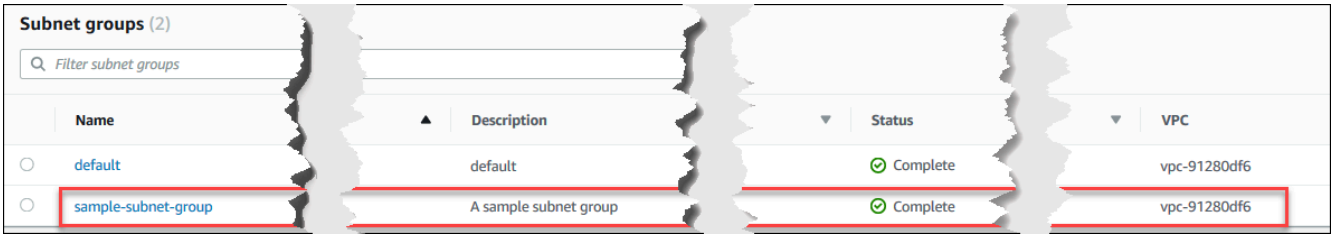
- Pour inclure tous les sous-réseaux dans le VPC sélectionné, choisissez Add all the subnets related to this VPC (Ajouter tous les sous-réseaux associés à ce VPC).
- Pour spécifier des sous-réseaux pour ce groupe de sous-réseaux, effectuez les actions suivantes pour chaque zone de disponibilité dans laquelle vous souhaitez inclure des sous-réseaux. Vous devez inclure au moins deux zones de disponibilité.

  A. Zone de disponibilité : dans la liste, choisissez une zone de disponibilité.

  B. Sous-réseau —Dans la liste, choisissez un sous-réseau dans la zone de disponibilité choisie pour ce groupe de sous-réseaux.

  C. Sélectionnez Add subnet (Ajouter le sous-réseau).

4. Sélectionnez Create (Créer). Lorsque le groupe de sous-réseaux est créé, il est répertorié avec vos autres groupes de sous-réseau.



Using the AWS CLI

Avant de pouvoir créer un groupe de sous-réseaux à l'aide de l'AWS CLI, vous devez d'abord identifier les sous-réseaux disponibles. Exécutez l'opération de l'AWS CLI suivante pour répertorier les zones de disponibilité et leurs sous-réseaux.

Paramètres :

- **`--db-subnet-group`**—Facultatif. Lorsque vous spécifiez un groupe de sous-réseaux en particulier, ses zones de disponibilité et ses sous-réseaux sont répertoriés. Si vous ne spécifiez pas ce paramètre, les zones de disponibilité et les sous-réseaux de tous vos groupes de sous-réseaux seront répertoriés. Si vous spécifiez le groupe de sous-réseaux `default`, tous les sous-réseaux du VPC sont répertoriés.

Example

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-subnet-groups \
    --db-subnet-group-name default \
    --query 'DBSubnetGroups[*].[DBSubnetGroupName,Subnets[*].
[SubnetAvailabilityZone.Name,SubnetIdentifier]]'
```

Pour Windows :

```
aws docdb describe-db-subnet-groups ^
    --db-subnet-group-name default ^
    --query 'DBSubnetGroups[*].[DBSubnetGroupName,Subnets[*].
[SubnetAvailabilityZone.Name,SubnetIdentifier]]'
```

La sortie de cette opération ressemble à ceci (format JSON).

```
[
    [
        "default",
        [
            [
                "us-east-1a",
                "subnet-4e26d263"
            ],
            [
                "us-east-1c",
                "subnet-afc329f4"
            ],
            [
                "us-east-1e",
                "subnet-b3806e8f"
            ],
            [
                "us-east-1d",
                "subnet-53ab3636"
            ],
            [
                "us-east-1b",
                "subnet-991cb8d0"
            ],
            [
                "us-east-1f",
                "subnet-29ab1025"
            ]
```

```
        ]
    ]
]
```

Avec le résultat de l'opération précédente, vous pouvez créer un groupe de sous-réseaux. Le nouveau groupe de sous-réseaux doit inclure des sous-réseaux d'au moins deux zones de disponibilité.

Paramètres :

- **`--db-subnet-group-name`** : obligatoire. Nom du groupe de sous-réseaux.
- **`--db-subnet-group-description`** : obligatoire. Description du groupe de sous-réseaux.
- **`--subnet-ids`** : obligatoire. Liste des sous-réseaux à inclure dans ce groupe de sous-réseaux. Exemple: `subnet-53ab3636`.
- --Tags —Facultatif. Liste de balises (paires clé-valeur) à associer à ce groupe de sous-réseaux.

Le code suivant crée le groupe de sous-réseaux `sample-subnet-group` avec trois sous-réseaux, `subnet-4e26d263`, `subnet-afc329f4` et `subnet-b3806e8f`.

Pour Linux, macOS ou Unix :

```
aws docdb create-db-subnet-group \
    --db-subnet-group-name sample-subnet-group \
    --db-subnet-group-description "A sample subnet group" \
    --subnet-ids subnet-4e26d263 subnet-afc329f4 subnet-b3806e8f \
    --tags Key=tag1,Value=One Key=tag2,Value=2
```

Pour Windows :

```
aws docdb create-db-subnet-group ^
    --db-subnet-group-name sample-subnet-group ^
    --db-subnet-group-description "A sample subnet group" ^
    --subnet-ids subnet-4e26d263 subnet-afc329f4 subnet-b3806e8f ^
    --tags Key=tag1,Value=One Key=tag2,Value=2
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
```

```
    "DBSubnetGroup": {
        "DBSubnetGroupDescription": "A sample subnet group",
        "DBSubnetGroupName": "sample-subnet-group",
        "Subnets": [
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1a"
                },
                "SubnetIdentifier": "subnet-4e26d263",
                "SubnetStatus": "Active"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1c"
                },
                "SubnetIdentifier": "subnet-afc329f4",
                "SubnetStatus": "Active"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1e"
                },
                "SubnetIdentifier": "subnet-b3806e8f",
                "SubnetStatus": "Active"
            }
        ],
        "VpcId": "vpc-91280df6",
        "DBSubnetGroupArn": "arn:aws:rds:us-east-1:123SAMPLE012:subgrp:sample-
subnet-group",
        "SubnetGroupStatus": "Complete"
    }
}
```

Description d'un groupe de sous-réseaux Amazon DocumentDB

Vous pouvez utiliser leAWS Management Console ou leAWS CLI pour obtenir les détails d'un groupe de sous-réseaux Amazon DocumentDB.

Using the AWS Management Console

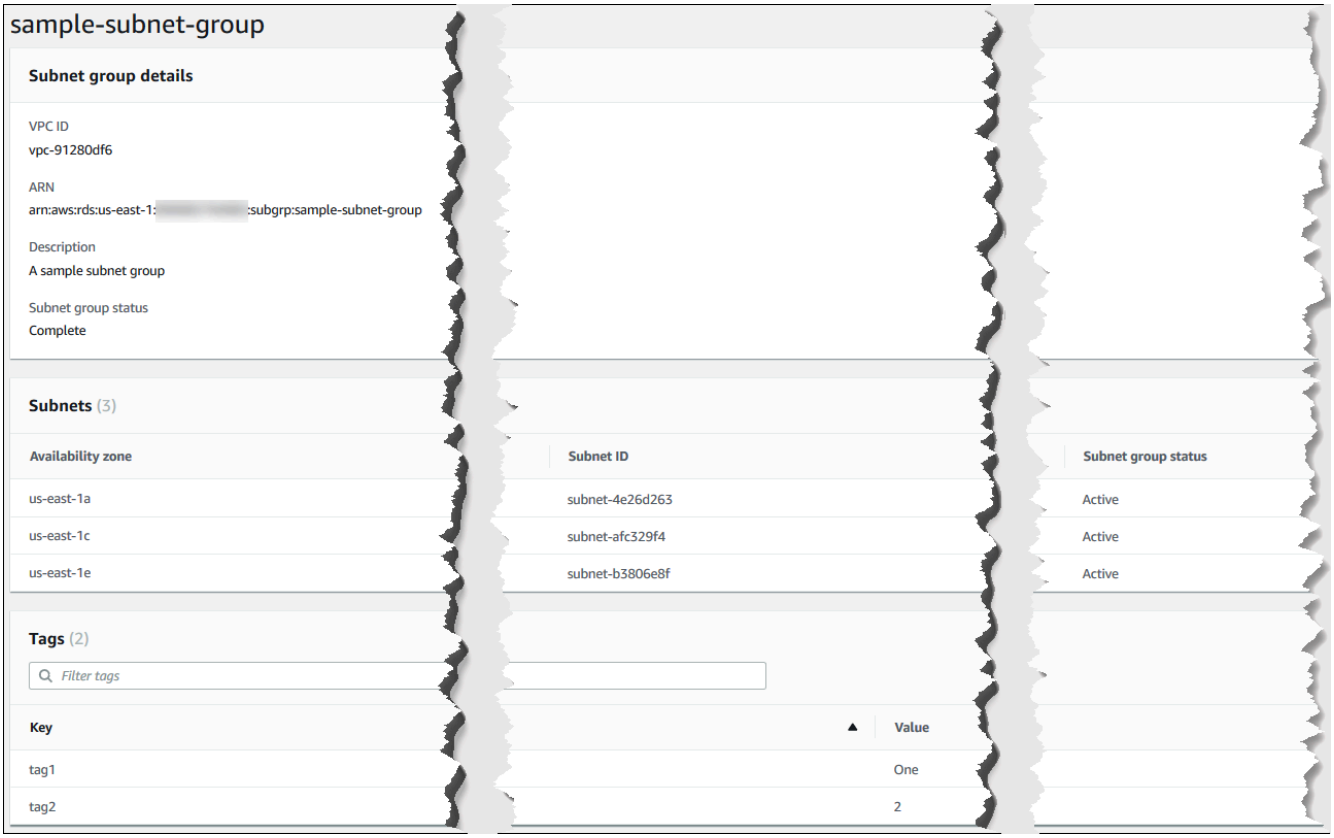
La procédure suivante vous montre comment obtenir des informations sur un groupe de sous-réseaux Amazon DocumentDB.

Pour trouver les informations d'un groupe de sous-réseaux

1. Connectez-vous à laAWS Management Console et ouvrez la console Amazon DocumentDB à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Subnet groups (Groupes de sous-réseaux).

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Pour afficher les informations d'un groupe de sous-réseaux, choisissez le nom du groupe.



Using the AWS CLI

Pour obtenir les détails d'un groupe de sous-réseaux Amazon DocumentDB, utilisez l'`describe-db-subnet-groups`opération avec le paramètre suivant.

Paramètre

- `--db-subnet=group-name`—Facultatif. Si inclues, les informations du groupe de sous-réseaux nommé sont répertoriées. Si omises, les informations d'un maximum de 100 groupes de sous-réseaux sont répertoriées.

Example

Le code suivant répertorie les informations du groupe de sous-réseaux `sample-subnet-group` que nous avons créé dans la section Création d'un groupe de sous-réseaux Amazon DocumentDB.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-subnet-groups \
    --db-subnet-group-name sample-subnet-group
```

Pour Windows :

```
aws docdb describe-db-subnet-groups ^
    --db-subnet-group-name sample-subnet-group
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupArn": "arn:aws:rds:us-east-1:123SAMPLE012:subgrp:sample-
subnet-group",
        "VpcId": "vpc-91280df6",
        "SubnetGroupStatus": "Complete",
        "DBSubnetGroupName": "sample-subnet-group",
        "Subnets": [
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1a"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-4e26d263"
            },
            {
```

```
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1c"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-afc329f4"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1e"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-b3806e8f"
            }
        ],
        "DBSubnetGroupDescription": "A sample subnet group"
    }
}
```

Modification d'un groupe de sous-réseaux Amazon DocumentDB

Vous pouvez utiliser leAWS Management Console ouAWS CLI pour modifier la description d'un groupe de sous-réseaux ou pour ajouter ou supprimer des sous-réseaux d'un groupe de sous-réseaux Amazon DocumentDB. Toutefois, vous ne pouvez pas modifier le groupe de sous-réseaux `default`.

Using the AWS Management Console

Vous pouvez utiliser la AWS Management Console pour modifier la description d’un groupe de sous-réseaux ou pour ajouter ou supprimer des sous-réseaux. Souvenez-vous que lorsque vous avez terminé, vous devez disposer d'au moins deux zones de disponibilité associées à votre groupe de sous-réseaux.

Pour modifier votre groupe de sous-réseaux

1. Connectez-vous à laAWS Management Console et ouvrez la console Amazon DocumentDB à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Subnet groups (Groupes de sous-réseaux). Choisissez ensuite le bouton situé à gauche du nom du groupe de sous-réseaux. Souvenez-vous que vous ne pouvez pas modifier le groupe de sous-réseaux `default`.

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Choisissez Actions, puis Modify (Modifier).
4. Description —Pour modifier la description de votre groupe de sous-réseaux, entrez une nouvelle description.
5. Pour modifier les sous-réseaux associés à votre groupe de sous-réseaux, effectuez l'une des actions suivantes dans la section Add subnets (Ajouter des sous-réseaux) :
   - Pour supprimer tous les sous-réseaux de ce groupe, choisissez Remove all (Supprimer tout).
   - Pour supprimer des sous-réseaux spécifiques de ce groupe, choisissez Remove (Supprimer) pour chaque sous-réseau à supprimer.
   - Pour ajouter tous les sous-réseaux associés à ce VPC, choisissez Add all the subnets related to this VPC (Ajouter tous les sous-réseaux associés à ce VPC).
   - Pour ajouter des sous-réseaux spécifiques à ce groupe, procédez comme suit pour chaque zone de disponibilité à laquelle vous souhaitez ajouter un sous-réseau.
     a. Zone de disponibilité : dans la liste, choisissez une nouvelle zone de disponibilité.
     b. Sous-réseau —Dans la liste, choisissez un sous-réseau dans la zone de disponibilité choisie pour ce groupe de sous-réseaux.
     c. Sélectionnez Add subnet (Ajouter le sous-réseau).
6. Dans la boîte de dialogue de confirmation :
   - Pour appliquer ces modifications au groupe de sous-réseaux, choisissez Modify (Modifier).
   - Pour ne pas les appliquer, choisissez Cancel (Annuler).

Using the AWS CLI

Vous pouvez utiliser la AWS CLI pour modifier la description d'un groupe de sous-réseaux ou pour ajouter ou supprimer des sous-réseaux. Souvenez-vous que lorsque vous avez terminé,

vous devez disposer d'au moins deux zones de disponibilité associées à votre groupe de sous-réseaux. Vous ne pouvez pas modifier le groupe de sous-réseaux `default`.

Paramètres :

- `--db-subnet-group-name` : obligatoire. Nom du groupe de sous-réseaux Amazon DocumentDB que vous modifiez.
- `--subnet-ids` : obligatoire. Liste de tous les sous-réseaux qui doivent être dans le groupe de sous-réseaux une fois que cette modification est effectuée.

  > **Important**
  >
  > Tous les sous-réseaux qui se trouvent actuellement dans le groupe de sous-réseaux et qui ne sont pas inclus dans cette liste seront supprimés du groupe. Si vous souhaitez conserver les sous-réseaux actuellement présents dans le groupe de sous-réseaux, incluez-les dans cette liste.

- `--db-subnet-group-description`—Facultatif. La description du groupe de sous-réseaux.

Example

Le code suivant modifie la description et remplace les sous-réseaux existants par les sous-réseaux `subnet-991cb8d0`, `subnet-53ab3636` et `subnet-29ab1025`.

Pour Linux, macOS ou Unix :

```
aws docdb modify-db-subnet-group \
    --db-subnet-group-name sample-subnet-group \
    --subnet-ids subnet-991cb8d0 subnet-53ab3636 subnet-29ab1025 \
    --db-subnet-group-description "Modified subnet group"
```

Pour Windows :

```
aws docdb modify-db-subnet-group ^
    --db-subnet-group-name sample-subnet-group ^
    --subnet-ids subnet-991cb8d0 subnet-53ab3636 subnet-29ab1025 ^
    --db-subnet-group-description "Modified subnet group"
```

La sortie de cette opération ressemble à ceci (format JSON). Notez qu'il s'agit du groupe de sous-réseaux qui a été créé dans la section Création d'un groupe de sous-réseaux Amazon

DocumentDB. Cependant, les sous-réseaux du groupe de sous-réseaux sont remplacés par ceux qui sont répertoriés dans l'opération `modify-db-subnet-group`.

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupArn": "arn:aws:rds:us-east-1:123SAMPLE012:subgrp:sample-
subnet-group",
        "DBSubnetGroupDescription": "Modified subnet group",
        "SubnetGroupStatus": "Complete",
        "Subnets": [
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1d"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-53ab3636"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1b"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-991cb8d0"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1f"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-29ab1025"
            }
        ],
        "VpcId": "vpc-91280df6",
        "DBSubnetGroupName": "sample-subnet-group"
    }
}
```

Suppression d'un groupe de sous-réseaux Amazon DocumentDB

Vous pouvez utiliser leAWS Management Console ouAWS CLI pour supprimer un groupe de sous-réseaux Amazon DocumentDB. Vous ne pouvez toutefois pas supprimer le groupe de sous-réseaux `default`.

Using the AWS Management Console

Vous pouvez utiliser la AWS Management Console pour supprimer un groupe de sous-réseaux. Mais vous ne pouvez pas supprimer le groupe de sous-réseaux `default`.

Pour supprimer un groupe de sous-réseaux

1. Connectez-vous à laAWS Management Console et ouvrez la console Amazon DocumentDB à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Subnet groups (Groupes de sous-réseaux). Choisissez ensuite le bouton situé à gauche du nom du groupe de sous-réseaux. Souvenez-vous que vous ne pouvez pas supprimer le groupe de sous-réseaux `default`.

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Choisissez Actions, puis Delete (Supprimer).
4. Dans la boîte de dialogue de confirmation :
   - Pour supprimer le groupe de sous-réseaux, choisissez Delete (Supprimer).
   - Pour conserver le groupe de sous-réseaux, choisissez Cancel (Annuler).

Using the AWS CLI

Pour supprimer un groupe de sous-réseaux Amazon DocumentDB à l'aide deAWS CLI, exécutez l'`delete-db-subnet-group`opération avec le paramètre suivant.

Paramètre

- `--db-subnet-group-name` : obligatoire. Nom du groupe de sous-réseaux Amazon DocumentDB à supprimer. Souvenez-vous que vous ne pouvez pas supprimer le groupe de sous-réseaux `default`.

Example

Le code suivant permet de supprimer `sample-subnet-group`.

Pour Linux, macOS ou Unix :

```
aws docdb delete-db-subnet-group \
    --db-subnet-group-name sample-subnet-group
```

Pour Windows :

```
aws docdb delete-db-subnet-group ^
    --db-subnet-group-name sample-subnet-group
```

L'opération ne produit aucun résultat.

Haute disponibilité et réplication Amazon DocumentDB

Vous pouvez obtenir une haute disponibilité et un dimensionnement en lecture dans Amazon DocumentDB (avec compatibilité MongoDB) en utilisant des instances de réplica. Un cluster Amazon DocumentDB prend en charge une seule instance principale et jusqu'à 15 instances de réplica. Ces instances peuvent être réparties sur les différentes zones de disponibilité au sein de la région du cluster. L'instance principale accepte le trafic en lecture et en écriture et les instances de réplica acceptent uniquement les demandes en lecture.

Le volume de cluster est composé de plusieurs copies des données du cluster. Cependant, les données du volume de cluster sont représentées comme un seul volume logique à l'instance principale et aux réplicas Amazon DocumentDB du cluster. Les instances de réplica sont cohérentes à terme. Elles renvoient les mêmes données pour les résultats des requêtes avec un retard de réplica minimal, généralement inférieur à 100 ms après l'écriture d'une mise à jour par l'instance principale. Le retard de réplica varie en fonction de la fréquence de modification de la base de

données. Autrement dit, pendant les périodes où une importante quantité d'opérations d'écriture se produit pour la base de données, il se peut que vous constatiez un retard accru du réplica.

Dimensionnement en lecture

Les réplicas Amazon DocumentDB fonctionnement parfaitement pour le dimensionnement en lecture, car ils sont intégralement dédiés aux opérations de lecture de votre volume de cluster. Les opérations d'écriture sont gérées par l'instance principale. Le volume du cluster est partagé entre toutes les instances de votre cluster. Par conséquent, vous n'avez pas besoin de répliquer et de conserver une copie des données pour chaque réplica Amazon DocumentDB.

Haute disponibilité

Lorsque vous créez un cluster Amazon DocumentDB, provisionne les instances dans les zones de disponibilité en fonction du nombre de ces zones dans le groupe de sous-réseaux (avec un minimum de deux). Lorsque vous créez des instances dans le cluster, Amazon DocumentDB répartit automatiquement les instances entre les zones de disponibilité d'un groupe de sous-réseaux afin d'équilibrer le cluster. Cette action évite également que toutes les instances soient situées dans la même zone de disponibilité.

Exemple

Pour illustrer cet exemple, imaginons que vous créez un cluster avec un groupe de sous-réseaux composé de trois zones de disponibilité : AZ1,AZ2, etAZ3.

Une fois la première instance du cluster créée, elle devient l'instance principale et elle est située dans l'une des zones de disponibilité. Dans cet exemple, il s'agit de la zone AZ1. La deuxième instance créée est une instance de réplica située dans l'une des deux autres zones de disponibilité, AZ2 dans notre exemple. La troisième instance créée est une instance de réplica située dans la zone de disponibilité restante, AZ3. Si vous créez plusieurs instances, elles sont réparties entre les zones de disponibilité afin d'équilibrer le cluster.

En cas de défaillance dans l'instance principale (AZ1), un basculement est déclenché et l'une des instances de réplica existantes est promue instance principale. Lorsque l'ancienne instance principale récupère, elle devient un réplica dans la zone de disponibilité dans laquelle elle a été mise en service (AZ1). Lorsque vous mettez en service un cluster à trois instances, Amazon DocumentDB continue de préserver ce cluster à trois instances. Amazon DocumentDB gère automatiquement la détection, le basculement et la récupération en cas d'échec des instances, sans qu'aucune intervention manuelle soit nécessaire.

Lorsque Amazon DocumentDB effectue un basculement et récupère une instance, l'instance récupérée reste dans la zone de disponibilité dans laquelle elle a été initialement mise en service. Toutefois, le rôle de l'instance peut changer, l'instance principale devenant instance de réplica. Cette opération permet d'éviter le scénario où une série de basculements entraîne la présence de toutes les instances dans la même zone de disponibilité.

Vous pouvez spécifier les réplicas Amazon DocumentDB comme cibles en cas de basculement. En d'autres termes, en cas de défaillance de l'instance principale, le réplica Amazon DocumentDB ou réplica d'un autre niveau est promu instance principale. Il y a une brève interruption, pendant laquelle les demandes de lecture et d'écriture adressées à l'instance principale échouent en renvoyant une exception. Si le cluster Amazon DocumentDB ne contient aucun réplica Amazon DocumentDB, l'instance principale est recréée pendant un échec. La promotion d'un réplica Amazon DocumentDB est beaucoup plus rapide que la recréation de l'instance principale.

Pour les scénarios de haute disponibilité, il est recommandé de créer un ou plusieurs réplicas Amazon DocumentDB. Ces réplicas doivent avoir la même classe d'instance que l'instance principale, et se trouver dans des zones de disponibilité différentes de votre cluster Amazon DocumentDB.

Pour plus d’informations, consultez les ressources suivantes :

- Comprendre la tolérance aux pannes des clusters Amazon DocumentDB
- Basculement Amazon DocumentDB
  - Contrôle de la cible du basculement

Haute disponibilité avec les clusters mondiaux

Pour une haute disponibilité sur plusieursRégions AWS, vous pouvez configurerClusters Amazon DocumentDB globaux. Un cluster global couvre plusieurs régions, ce qui assure une faible latence des lectures globales et la reprise après sinistre en cas de panne dans unRégion AWS. Amazon DocumentDB gère automatiquement la réplication de toutes les données et mises à jour de la région principale vers chacune des régions secondaires.

Ajout de réplicas

La première instance ajoutée au cluster est l'instance principale. Chaque instance ajoutée après la première instance est une instance de réplica. Un cluster peut avoir jusqu'à 15 instances de réplica en plus de l'instance principale.

Lorsque vous utilisez la AWS Management Console pour créer un cluster, une instance principale est automatiquement créée en même temps. Pour créer un réplica en même temps que vous créez le cluster et l'instance principale, choisissez Créer un réplica dans différentes zones. Pour plus d'informations, consultez l'étape 4.d dans Création d'un cluster Amazon DocumentDB. Pour ajouter d'autres réplicas à un cluster Amazon DocumentDB, veuillez consulterAjouter une instance Amazon DocumentDB à un cluster.

Si vous utilisez la AWS CLI pour créer votre cluster , vous devez créer explicitement vos instances principale et de réplica. Pour de plus amples informations, veuillez consulter la section Utilisation de l'AWS CLI dans les rubrique suivantes :

- Création d'un cluster Amazon DocumentDB
- Ajouter une instance Amazon DocumentDB à un cluster

Basculement Amazon DocumentDB

Amazon DocumentDB (avec compatibilité MongoDB) détecte l'échec et remplace le nœud principal. Au cours d'un basculement, le délai d'écriture est réduit. En effet, le rôle du nœud principal est transféré à l'un des réplicas en lecture et il n'est pas nécessaire de créer et d'allouer un nouveau nœud principal. La détection d'un échec ou la promotion d'un réplica vous permettent de recommencer à écrire dans le nouveau nœud principal dès que la promotion est terminée.

Pour que le basculement fonctionne, votre cluster doit avoir au moins deux instances : une instance principale et au moins une réplica.

Contrôle de la cible du basculement

Amazon DocumentDB fournit des niveaux de basculement comme moyens de contrôler quel réplica d'instance est promu en instance principale lorsqu'un basculement se produit.

Niveaux de basculement

Chaque instance de réplica est associée à un niveau de basculement (0—15). Lorsqu'un basculement se produit en raison de la maintenance ou d'une improbable panne matérielle, l'instance principale bascule vers un réplica avec la priorité la plus élevée (niveau le plus bas numéro). Si plusieurs réplicas ont le même niveau de priorité, le principal bascule vers ce niveau de réplica qui est le plus proche de la taille du principal précédent.

En définissant le niveau de basculement pour un groupe de réplicas sélectionnés pour `0` (la priorité la plus haute), vous pouvez vous assurer qu'un basculement promeut l'un des réplicas dans ce groupe. Une manière efficace d'empêcher que des réplicas spécifiques ne soient promus en principal en cas de basculement consiste à leur attribuer un niveau de priorité faible (chiffre élevé). Cela s'avère utile dans les cas où les réplicas spécifiques font l'objet d'une utilisation intensive de la part d'une application et le basculement vers l'un d'entre eux pourrait avoir un impact négatif sur une application critique.

Vous pouvez définir le niveau de basculement d'une instance lorsque vous la créez ou plus tard en la modifiant. Le définition d'un niveau de basculement d'une instance en modifiant cette dernière ne déclenche pas un basculement. Pour plus d'informations, consultez les rubriques suivantes :

- Ajouter une instance Amazon DocumentDB à un cluster
- Modification d'une instance Amazon DocumentDB

Lorsque vous lancez manuellement un basculement, vous avez deux moyens de contrôler quelle instance de réplica est promue en instance principale : le niveau de basculement comme décrit précédemment, et le paramètre `--target-db-instance-identifier`.

`--target-db-instance-identifier`

Pour le test, vous pouvez forcer un événement de basculement à l'aide de l'opération `failover-db-cluster`. Vous pouvez utiliser le paramètre `--target-db-instance-identifier` pour spécifier le réplica à promouvoir en réplica principal. L'utilisation du paramètre `--target-db-instance-identifier` remplace le niveau de priorité de basculement. Si vous ne spécifiez pas le paramètre `--target-db-instance-identifier`, le basculement principal est conforme au niveau de priorité de basculement.

Que se passe-t-il pendant un basculement ?

Amazon DocumentDB gère automatiquement le basculement, afin que vos applications puissent reprendre vos opérations de base de données aussi vite que possible sans intervention administrative.

- Si vous avez une instance de réplica Amazon DocumentDB dans la même zone de disponibilité ou une zone de disponibilité différente lorsque vous échouez : Amazon DocumentDB retourne l'enregistrement de nom canonique (CNAME) de votre instance afin qu'il pointe vers le réplica

sain, qui est choisi à son tour pour devenir la nouvelle instance principale. Le basculement complet s'effectue généralement en 30 secondes.

- Si vous n'avez pas d'instance Amazon DocumentDB de réplica (par exemple, un cluster d'instance unique) : Amazon DocumentDB tente de créer une nouvelle instance dans la même zone de disponibilité que l'instance d'origine. Ce remplacement de l'instance d'origine s'effectue de manière optimale et peut échouer si, par exemple, un problème affecte de manière générale la zone de disponibilité.

Votre application devrait tenter une nouvelle connexion à la base de données dans le cas d'une perte de connexion.

Test du basculement

Un basculement pour un cluster promeut l'un des réplica Amazon DocumentDB (instances en lecture seule) du cluster en instance principale (enregistreur du cluster).

Si l'instance principale échoue, Amazon DocumentDB bascule automatiquement vers un réplica Amazon DocumentDB, le cas échéant. Vous pouvez forcer un basculement lorsque vous souhaitez simuler un échec d'une instance principale à des fins de test. Chaque instance d'un cluster possède sa propre adresse de point de terminaison. Vous devez donc nettoyer et rétablir toutes les connexions existantes qui utilisent ces adresses de point de terminaison lorsque le basculement est effectué.

Pour forcer un basculement, utilisez l'opération `failover-db-cluster` avec ces paramètres.

- `--db-cluster-identifier` : obligatoire. Nom du cluster qui doit basculer.
- `--target-db-instance-identifier`—Facultatif. Le nom de l'instance à promouvoir comme instance principale.

Example

L'opération suivante force un basculement du cluster `sample-cluster`. Cela ne spécifie pas l'instance qui doit devenir la nouvelle instance principale, par conséquent, Amazon DocumentDB choisit l'instance en fonction du niveau de priorité de basculement.

Pour Linux, macOS ou Unix :

```
aws docdb failover-db-cluster \
```

```
    --db-cluster-identifier sample-cluster
```

Pour Windows :

```
aws docdb failover-db-cluster ^
    --db-cluster-identifier sample-cluster
```

L'opération suivante force un basculement du cluster `sample-cluster`, en spécifiant quelle `sample-cluster-instance` est à promouvoir en tant que rôle principal. (Notez la valeur `"IsClusterWriter": true` dans la sortie).

Pour Linux, macOS ou Unix :

```
aws docdb failover-db-cluster \
    --db-cluster-identifier sample-cluster \
    --target-db-instance-identifier sample-cluster-instance
```

Pour Windows :

```
aws docdb failover-db-cluster ^
    --db-cluster-identifier sample-cluster ^
    --target-db-instance-identifier sample-cluster-instance
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "DBCluster": {
        "HostedZoneId": "Z2SUY0A1719RZT",
        "Port": 27017,
        "EngineVersion": "3.6.0",
        "PreferredMaintenanceWindow": "thu:04:05-thu:04:35",
        "BackupRetentionPeriod": 1,
        "ClusterCreateTime": "2018-06-28T18:53:29.455Z",
        "AssociatedRoles": [],
        "DBSubnetGroup": "default",
        "MasterUsername": "master-user",
        "Engine": "docdb",
        "ReadReplicaIdentifiers": [],
        "EarliestRestorableTime": "2018-08-21T00:04:10.546Z",
        "DBClusterIdentifier": "sample-cluster",
        "ReaderEndpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "DBClusterMembers": [
```

```
            {
                "DBInstanceIdentifier": "sample-cluster-instance",
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1,
                "IsClusterWriter": true
            },
            {
                "DBInstanceIdentifier": "sample-cluster-instance-00",
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1,
                "IsClusterWriter": false
            },
            {
                "DBInstanceIdentifier": "sample-cluster-instance-01",
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1,
                "IsClusterWriter": false
            }
        ],
        "AvailabilityZones": [
            "us-east-1b",
            "us-east-1c",
            "us-east-1a"
        ],
        "DBClusterParameterGroup": "default.docdb3.6",
        "Endpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "IAMDatabaseAuthenticationEnabled": false,
        "AllocatedStorage": 1,
        "LatestRestorableTime": "2018-08-22T21:57:33.904Z",
        "PreferredBackupWindow": "00:00-00:30",
        "StorageEncrypted": false,
        "MultiAZ": true,
        "Status": "available",
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-12345678"
            }
        ],
        "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    }
}
```

Temps de réplication

Le décalage de réplication est généralement de 50 ms ou moins. Les raisons les plus courantes de l'augmentation du retard de réplica sont les suivantes :

- Un taux d'écriture élevé sur le principal qui entraîne le retard des réplicas en lecture par rapport au principal.
- Contestation sur les réplicas en lecture entre requêtes longues (par exemple, analyses séquentielles volumineuses, requêtes d'agrégation) et la réplication en écriture entrante.
- Très grand nombre de requêtes simultanées sur les réplicas en lecture.

Pour minimiser le décalage de réplication, essayez les techniques de dépannage suivantes :

- Si vous avez un taux d'écriture élevé ou une utilisation élevée du processeur, nous vous recommandons de faire évoluer les instances de votre cluster.
- S'il existe des requêtes longues sur vos réplicas en lecture et que des mises à jour très fréquentes des documents interrogés sont effectuées, envisagez de modifier vos requêtes longues ou de les exécuter sur le réplica principal/écriture pour éviter toute contestation sur les réplicas en lecture.
- S'il existe un très grand nombre de requêtes simultanées ou une utilisation élevée du processeur uniquement sur les réplicas en lecture, une autre option consiste à augmenter le nombre de réplicas en lecture pour étaler la charge de travail.
- Étant donné que le décalage de réplication est le résultat d'un débit d'écriture élevé et de requêtes longues, nous vous recommandons de résoudre le retard de réplication en utilisant la métrique DBClusterReplicaIAGMaximum CW en combinaison avec l'enregistreur de requêtes lentes et`WriteThroughput`/`WriteIOPS`métriques ,

En général, nous recommandons que tous vos réplicas soient du même type d'instance, de sorte qu'un basculement de cluster ne provoque pas de dégradation des performances.

Si vous choisissez entre la mise à l'échelle et la mise à l'échelle (par exemple, six instances plus petites contre trois instances plus grandes), nous vous recommandons généralement d'essayer d'effectuer une mise à l'échelle (instances plus grandes) avant d'effectuer une mise à l'échelle, car vous obtiendrez un cache tampon plus important par instance DB.

De manière proactive, vous devez définir une alarme de décalage de réplication et définir son seuil sur une valeur qui, selon vous, est la limite supérieure pour déterminer la distance derrière (ou « périmée ») de vos données sur les instances de réplica avant qu'elles ne commencent à affecter

les fonctionnalités de votre application. En général, nous recommandons que le seuil de retard de réplication soit dépassé pour plusieurs points de données avant d'être alarmé, en raison de charges de travail transitoires.

> **Note**
>
> En outre, nous vous recommandons de définir une autre alarme pour les retards de réplication qui dépassent 10 secondes. Si vous dépassez ce seuil pour plusieurs points de données, nous vous recommandons d'augmenter vos instances ou de réduire votre débit d'écriture sur l'instance principale.

Gestion des index Amazon DocumentDB

Création d'index Amazon DocumentDB

La création d'index dans Amazon DocumentDB nécessite la prise d'un certain nombre de décisions :

- Dans quel délai doit-il être terminé ?
- La collection peut-elle être inaccessible pendant la construction ?
- Quelle quantité de puissance de calcul d'une instance peut être allouée à la construction ?
- Quel type d'index faut-il créer ?

Cette section vous aide à répondre à ces questions et fournit les commandes et les exemples de surveillance pour créer un index Amazon DocumentDB sur votre collection de clusters basée sur une instance.

Consignes

Les directives suivantes incluent les limites de base et les compromis de configuration lors de la création de nouveaux index :

- Prise en charge des versions d'Amazon DocumentDB : alors que l'indexation par un seul travailleur est prise en charge sur toutes les versions d'Amazon DocumentDB, l'indexation par plusieurs travailleurs n'est prise en charge que sur les versions 4.0 et 5.0 d'Amazon DocumentDB.
- Compromis en termes de performances : l'augmentation du nombre de travailleurs participant au processus de création de l'index augmente l'utilisation du processeur et augmente les E/S de

lecture sur l'instance principale de votre base de données Amazon DocumentDB. Les ressources nécessaires à la création d'un nouvel index ne seront pas disponibles pour votre charge de travail en cours d'exécution.

- Clusters élastiques : l'indexation parallèle n'est pas prise en charge sur les clusters élastiques Amazon DocumentDB.
- Nombre maximum de travailleurs : le nombre maximum de travailleurs que vous pouvez configurer dépend de la taille de votre instance principale dans votre cluster de base de données. Cela représente la moitié du nombre total de vCPU sur l'instance principale de votre cluster de base de données. Par exemple, vous pouvez exécuter un maximum de 32 travailleurs sur une instance db.r6g.16xlarge dotée de 64 vCPU.

> **Note**
>
> Les travailleurs parallèles ne sont pas pris en charge sur les classes d'instance 2xlarge et inférieures.

- Nombre minimum de travailleurs : le nombre minimum de travailleurs que vous pouvez configurer est de un. Le paramètre par défaut pour la création d'index sur des clusters basés sur des instances est de deux travailleurs. Cependant, vous pouvez réduire le nombre de travailleurs à un en utilisant l'option « threads de travail ». Cela exécutera le processus avec un seul travailleur.
- Compression d'index : Amazon DocumentDB ne prend pas en charge la compression d'index. La taille des données pour les index peut être plus importante que lorsque vous utilisez d'autres options.
- Indexation de plusieurs collections : la moitié des vCPU de l'instance principale de votre cluster de base de données peuvent être utilisées par des opérateurs configurés qui créent des index sur plusieurs collections.
- Types d'index : consultez ce billet de blog pour obtenir une explication complète des types d'index pris en charge sur Amazon DocumentDB.

Premiers pas

Pour démarrer la création d'index sur une collection, utilisez la `createIndexes` commande. Par défaut, la commande exécute deux tâches parallèles, ce qui multiplie par deux la vitesse du processus de création d'index.

Par exemple, le processus de commande suivant montre comment créer un index pour le champ « user_name » dans un document et augmenter la vitesse du processus d'indexation à quatre travailleurs :

1. Créez des index à l'aide de deux workers parallèles sur le cluster :

```
db.runCommand({"createIndexes":"test","indexes":[{"key": {"user_name":1},
 "name":"username_idx"}]})
```

2. Pour optimiser la vitesse du processus de création d'index, vous pouvez spécifier le nombre de travailleurs en utilisant l'option « threads de travail » (`"workers":<number>`) dans la `db.runCommand createIndexes` commande.

   Augmentez la vitesse du processus à quatre travailleurs parallèles :

```
db.runCommand({"createIndexes":"test","indexes":[{"key": {"user_name":1},
 "name":"username_idx", "workers":4}]})
```

   > **Note**
   >
   > Plus le nombre de travailleurs est élevé, plus la création de l'indice progresse rapidement. Toutefois, plus le nombre de travailleurs augmente, plus la charge sur les vCPU et les E/S de lecture de votre instance principale augmente. Assurez-vous que votre cluster est suffisamment provisionné pour faire face à la charge accrue sans dégrader les autres charges de travail.

État de progression de l'indexation

Le processus de création d'index fonctionne en initialisant, en scannant les collections, en triant les clés et, enfin, en insérant des clés au moyen d'un générateur d'index. Le processus comporte jusqu'à six étapes lorsque vous l'exécutez au premier plan, et jusqu'à neuf étapes lorsque vous l'exécutez en arrière-plan. Vous pouvez consulter les indicateurs d'état tels que le pourcentage d'achèvement, le nombre total de blocs de stockage numérisés, les clés triées et les clés insérées étape par étape.

Surveillez la progression du processus d'indexation à l'aide de la `db.currentOp()` commande dans le shell mongo. L'achèvement à 100 % de la dernière étape indique que tous les index ont été créés avec succès :

```
db.currentOp({"command.createIndexes": { $exists : true } })
```

Types de construction d'index

Les quatre types de constructions d'index sont les suivants :

- Premier plan : la version de l'index de premier plan bloque toutes les autres opérations de base de données jusqu'à ce que l'index soit créé. La version de premier plan d'Amazon DocumentDB comprend cinq étapes.
- Premier plan (unique) - Les versions d'index de premier plan pour un seul document (unique) bloquent d'autres opérations de base de données, comme les versions de premier plan classiques. Contrairement à la version de premier plan de base, la version unique utilise une étape supplémentaire (tri des clés 2) pour rechercher les clés dupliquées. La version de premier plan (unique) comprend six étapes.
- Arrière-plan : la génération de l'index en arrière-plan permet à d'autres opérations de base de données de s'exécuter au premier plan pendant la création de l'index. La construction en arrière-plan d'Amazon DocumentDB comprend huit étapes.
- Arrière-plan (unique) - Les versions d'index d'arrière-plan pour un seul document (unique) permettent à d'autres opérations de base de données de s'exécuter au premier plan pendant la création de l'index. Contrairement à la version d'arrière-plan de base, la version unique utilise une étape supplémentaire (tri des clés 2) pour rechercher les clés dupliquées. La version d'arrière-plan (unique) comprend neuf étapes.

Étapes de création de l'index

Étape	Premier plan	Premier plan (unique)	Contexte	Contexte (unique)
Initialisation	1	1	1	1
index de construction : initialisation	2	2	2	2
index des bâtiments :	3	3	3	3

Étape	Premier plan	Premier plan (unique)	Contexte	Contexte (unique)
collection de numérisation				
index du bâtiment : clés de tri 1	4	4	4	4
index du bâtiment : clés de tri 2		5		5
index de construction : insertion de clés	5	6	5	6
validation : index de numérisation			6	7
validation : tri des tuples			7	8
validation : numérisation de la collection			8	9

- initialisation - CreateIndex prépare le générateur d'index. Cette phase doit être très brève.
- building index : initialization - Le générateur d'index se prépare à créer l'index. Cette phase doit être très brève.
- index de construction : analyse de la collection - Le générateur d'index effectue une analyse de collection pour collecter les clés d'index. L'unité de mesure est le « bloc ».

> **Note**
>
> Si plusieurs outils de travail sont configurés pour la création de l'index, ils sont affichés à cette étape. L'étape de « numérisation de la collecte » est la seule étape qui utilise

plusieurs outils de traitement pendant le processus de création de l'index. Toutes les autres étapes afficheront un seul travailleur.

- index de construction : tri des clés 1 - Le générateur d'index trie les clés d'index collectées. L'unité de mesure est « clés ».
- index de construction : clés de tri 2 - Le générateur d'index trie les clés d'index collectées qui correspondent à des tuples morts. Cette phase n'existe que pour la création d'index uniques. L'unité de mesure est « clés ».
- index de construction : insertion de clés - Le générateur d'index insère des clés d'index dans le nouvel index. L'unité de mesure est « clés ».
- validation : index de numérisation - CreateIndex analyse l'index pour trouver les clés qui doivent être validées. L'unité de mesure est le « bloc ».
- validation : tri des tuples - CreateIndex trie le résultat de la phase de numérisation de l'index.
- validation : numérisation de la collection - CreateIndex analyse la collection pour valider les clés d'index trouvées lors des deux phases précédentes. L'unité de mesure est le « bloc ».

Exemple de sortie de création d'index

Dans l'exemple de sortie ci-dessous (création d'index de premier plan), l'état de la création de l'index est affiché. Le champ « msg » résume la progression de la construction en indiquant l'étape et le pourcentage d'achèvement de la construction. Le champ « travailleurs » indique le nombre de travailleurs utilisés au cours de cette étape de la construction de l'indice. Le champ « progression » indique les chiffres réels utilisés pour calculer le pourcentage d'achèvement.

Note

Les champs « currentIndexBuild Nom », « msg » et « progression » ne sont pas pris en charge sur Amazon DocumentDB version 4.0.

```
{
    "inprog" : [{
    …
        "command": {
            "createIndexes": "test",
            "indexes": [{
```

```
                "v": 2,
                "key": {
                    "user_name": 1
                },
                "name": "user_name_1"
            }],
            "lsid": {
                "id": UUID(“094d0fba-8f41-4373-82c3-7c4c7b5ff13b”)
            },
            "$db": "test"
        },
        "currentIndexBuildName": user_name_1,
        "msg": "Index Build: building index number_1, stage 6/6 building index:
 656860/1003520 (keys) 65%",
        "workers": 1,
        "progress": {
            "done": 656861,
            "total": 1003520
        },
    …
    ],

    "ok" : 1
}
```

Gestion de la compression des documents au niveau de la collection

La compression de documents au niveau de la collection Amazon DocumentDB vous permet de réduire les coûts de stockage et d'E/S en compressant les documents de vos collections. Vous pouvez activer la compression des documents au niveau de la collection et afficher les mesures de compression selon vos besoins en mesurant les gains de stockage grâce à des mesures de compression telles que la taille de stockage des documents compressés et l'état de compression. Amazon DocumentDB utilise l'algorithme de compression LZ4 pour compresser des documents.

Consignes

Les directives suivantes s'appliquent à la compression de documents au niveau de la collection :

- La compression des documents est désactivée par défaut

- La compression de documents ne peut pas être appliquée à des collections existantes.
- La compression de documents est uniquement prise en charge sur Amazon DocumentDB version 5.0 et ultérieure.
- Amazon DocumentDB compresse uniquement les documents dont la taille est supérieure ou égale à 2 Ko.

Activation de la compression de documents

Activez la compression de documents lors de la création d'une collection sur Amazon DocumentDB en utilisant la `db.createCollection()` méthode suivante :

```
db.createCollection( sample_collection,{
    storageEngine : {
        documentDB: {
            compression:{
                enable: <true | false>
            }
        }
    }
})
```

Surveillance de la compression des documents

Vous pouvez vérifier si une collection est compressée et calculer son taux de compression comme suit.

Consultez les statistiques de compression en exécutant la `db.collection.stats()` commande `db.printCollectionStats()` or depuis le shell mongo. La sortie indique la taille d'origine et la taille compressée que vous pouvez comparer pour analyser les gains de stockage résultant de la compression des documents. Dans cet exemple, les statistiques d'une collection nommée « sample_collection » sont affichées :

```
db.sample_collection.stats(1024*1024)

{
    "ns" : "test.sample_collection",
    "count" : 1000000,
    "size" : 3906.3,
    "avgObjSize" : 4096,
```

```
    "storageSize" : 1953.1,
    compression:{
        "enabled" : true,
        "threshold" : 2032
    }
    ...
}
```

- taille : taille d'origine de la collection de documents.
- avgObjSize- Taille moyenne du document avant compression arrondie à la première décimale. L'unité de mesure est l'octet.
- StorageSize : taille de stockage de la collection après compression. L'unité de mesure est l'octet.
- activé : indique si la compression est activée ou désactivée.

Pour calculer le taux de compression réel, divisez la taille de la collection par la taille de stockage (Size/StorageSize). Dans l'exemple ci-dessus, le calcul est 3906,3/1953,1, ce qui se traduit par un taux de compression de 2:1.

Gestion des collections existantes

Bien que vous ne puissiez pas compresser une collection existante, vous pouvez convertir des documents compressés ou non compressés. Pour stocker des documents non compressés existants au format compressé, copiez le document dans une collection compatible avec la compression. Pour convertir des documents compressés au format non compressé, copiez-les dans une collection dont la compression est désactivée.

Gestion des événements Amazon DocumentDB mentents

Amazon DocumentDB mentents (avec compatibilité avec Mongoents) consients les événements associés à vos clusters, instances, snapents, groupes de sécurité et groupes de paramètres de cluster. Les informations consignées comprennent la date et l'heure de l'événement, le nom et le type de source de l'événement, ainsi qu'un message associé à l'événement.

> **Important**
>
> Pour certaines fonctionnalités de gestion, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon RDS et Amazon Neptune. Les limites régionales, qui

sont régies au niveau de la région, sont partagées entre Amazon DocumentDB, Amazon RDS et Amazon Neptune. Pour plus d'informations, veuillez consulter Quotas régionaux.

Rubriques

- Affichage des catégories Amazon DocumentDB mentents
- Affichage des événements Amazon DocumentDB mentents

Affichage des catégories Amazon DocumentDB mentents

Chaque type de ressource Amazon DocumentDB possède des types d'événements spécifiques qui peuvent lui être associés. Vous pouvez utiliser cetteAWS CLI`describe-event-categories` opération pour visualiser le mappage entre les types d'événements et les types de ressources Amazon DocumentDB.

Paramètres

- **`--source-type`**—Facultatif. Utilisez le paramètre `--source-type` pour afficher les catégories d'événement d'un type source en particulier. Les valeurs suivantes sont autorisées :
  - `db-cluster`
  - `db-instance`
  - `db-parameter-group`
  - `db-security-group`
  - `db-cluster-snapshot`
- **`--filters`**—Facultatif. Pour afficher les catégories d'événements uniquement pour Amazon DocumentDB, utilisez le filtre`--filter Name=engine,Values=docdb`.

Example

Le code suivant répertorie les catégories d'événement associées à des clusters.

Pour Linux, macOS ou Unix :

```
aws docdb describe-event-categories \
    --filter Name=engine,Values=docdb \
    --source-type db-cluster
```

Pour Windows :

```
aws docdb describe-event-categories ^
    --filter Name=engine,Values=docdb ^
    --source-type db-cluster
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "EventCategoriesMapList": [
        {
            "EventCategories": [
                "notification",
                "failure",
                "maintenance",
                "failover"
            ],
            "SourceType": "db-cluster"
        }
    ]
}
```

Le code suivant répertorie les catégories d'événements associées à chaque type de source Amazon DocumentDB.

```
aws docdb describe-event-categories
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "EventCategoriesMapList": [
        {
            "SourceType": "db-instance",
            "EventCategories": [
                "notification",
                "failure",
                "creation",
                "maintenance",
                "deletion",
                "recovery",
                "restoration",
                "configuration change",
```

```
                "read replica",
                "backtrack",
                "low storage",
                "backup",
                "availability",
                "failover"
            ]
        },
        {
            "SourceType": "db-security-group",
            "EventCategories": [
                "configuration change",
                "failure"
            ]
        },
        {
            "SourceType": "db-parameter-group",
            "EventCategories": [
                "configuration change"
            ]
        },
        {
            "SourceType": "db-cluster",
            "EventCategories": [
                "notification",
                "failure",
                "maintenance",
                "failover"
            ]
        },
        {
            "SourceType": "db-cluster-snapshot",
            "EventCategories": [
                "backup"
            ]
        }
    ]
}
```

Affichage des événements Amazon DocumentDB mentents

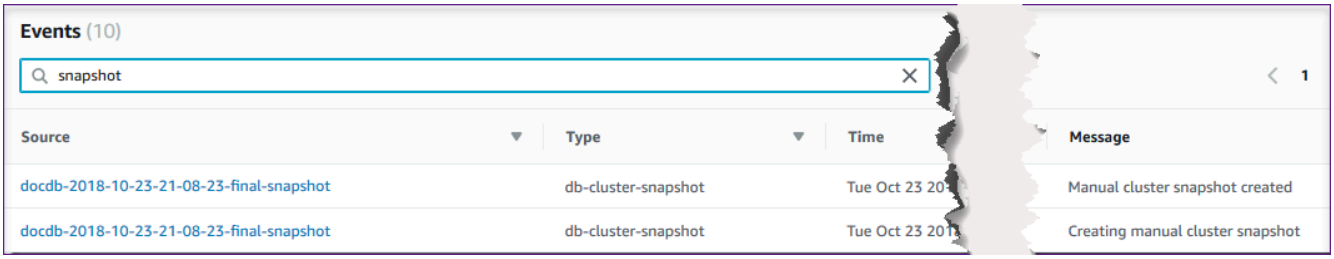
Vous pouvez récupérer les événements pour vos ressources Amazon DocumentDB Documentents ents mentents via la Amazon DocumentDB mentents, qui affiche les événements des dernières

24 heures. Vous pouvez également récupérer des événements pour vos ressources Amazon DocumentDB Documentents ents à l'aide de la commande de l', à l'aide de laAWS CLI commande de l', ou de l'opération d'API DescribeEventsAmazon mentents. Si vous utilisez l'AWS CLIou l'API Amazon DocumentDB pour afficher les événements, vous pouvez récupérer les événements datant de 14 derniers jours.

Using the AWS Management Console

Pour afficher tous les événements des instances Amazon DocumentDB mentents dernières

1. Connectez-vous à laAWS Management Console et ouvrez la console Amazon DocumentDB l'adresse https://console.aws.amazon.com/docdb.
2. Dans le volet de navigation, sélectionnez Events. Les évènements disponibles s'affichent sous forme de liste.
3. Utilisez la liste Filtre pour filtrer les événements par type. Saisissez un terme dans la zone de texte pour affiner vos résultats. Par exemple, la capture d'écran suivante montre le filtrage de tous les événements Amazon DocumentDB Documentents ents mentents.



Using the AWS CLI

Pour afficher tous les événements des instances Amazon mentents des 7 derniers jours

Vous pouvez afficher tous les événements des Amazon DocumentDB Documentents ents ents des 7 derniers jours en exécutant l'AWS CLIopération de l', et en définissant le`--duration` paramètre sur`10080` (10 080 minutes).

```
aws docdb describe-events --duration 10080
```

Filtrage des événements Amazon DocumentDB

Pour voir des événements Amazon DocumentDB spécifiques, utilisez l'`describe-events`opération avec les paramètres suivants.

Paramètres

- **`--filter`**—Obligatoire pour limiter les valeurs renvoyées aux événements Amazon DocumentDB. Permet**`Name=engine,Values=docdb`** de filtrer tous les événements pour Amazon DocumentDB uniquement.
- **`--source-identifier`**—Facultatif. Identifiant de la source de l'événement pour laquelle les événements sont renvoyés. Si cet argument n'est pas spécifié, les événements de toutes les sources sont inclus dans les résultats.
- **`--source-type`**—Facultatif, sauf si cela`--source-identifier` est fourni, alors obligatoire. Si le `--source-identifier` est fourni, le `--source-type` doit approuver le type du `--source-identifier`. Les valeurs suivantes sont autorisées :
  - `db-cluster`
  - `db-instance`
  - `db-parameter-group`
  - `db-security-group`
  - `db-cluster-snapshot`

L'exemple suivant répertorents tous vos événements Amazon DocumentDB ents ents.

```
aws docdb describe-events --filters Name=engine,Values=docdb
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "Events": [
        {
            "SourceArn": "arn:aws:rds:us-east-1:123SAMPLE012:db:sample-cluster-
instance3",
            "Message": "instance created",
            "SourceType": "db-instance",
            "Date": "2018-12-11T21:17:40.023Z",
            "SourceIdentifier": "sample-cluster-instance3",
            "EventCategories": [
                "creation"
            ]
        },
        {
```

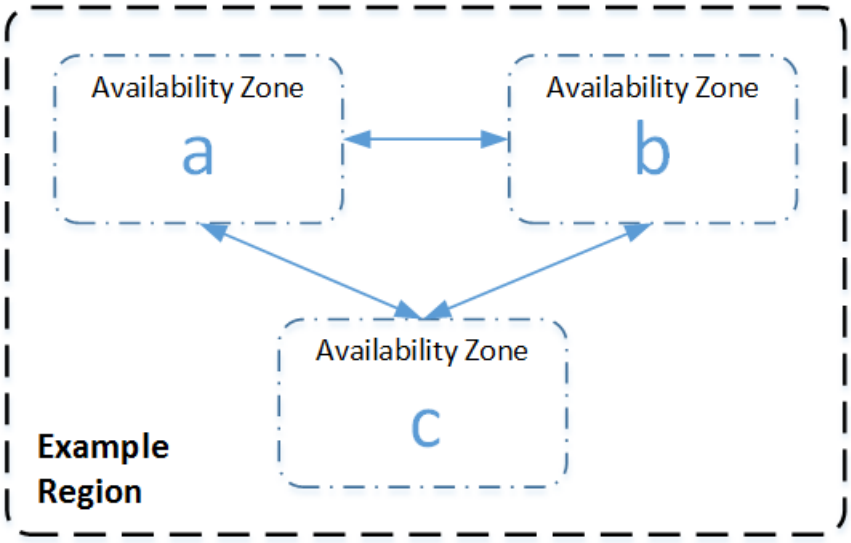
```
            "SourceArn": "arn:aws:rds:us-
east-1:123SAMPLE012:db:docdb-2018-12-11-21-08-23",
            "Message": "instance shutdown",
            "SourceType": "db-instance",
            "Date": "2018-12-11T21:25:01.245Z",
            "SourceIdentifier": "docdb-2018-12-11-21-08-23",
            "EventCategories": [
                "availability"
            ]
        },
        {
            "SourceArn": "arn:aws:rds:us-
east-1:123SAMPLE012:db:docdb-2018-12-11-21-08-23",
            "Message": "instance restarted",
            "SourceType": "db-instance",
            "Date": "2018-12-11T21:25:11.441Z",
            "SourceIdentifier": "docdb-2018-12-11-21-08-23",
            "EventCategories": [
                "availability"
            ]
        }
    ]
}
```

Pour plus d'informations, veuillez consulter Audit des événements Amazon DocumentDB.

Choix des régions et zones de disponibilité

Les ressources de cloud computing Amazon sont hébergées dans plusieurs emplacements à travers le monde. Ces emplacements se composent Régions AWS de zones de disponibilité. Chacune Région AWSconstitue une zone géographique distincte. Chaque région se compose de plusieurs emplacements isolés appelés zones de disponibilité. Amazon DocumentDB vous permet de placer des ressources, telles que des instances, et des données dans plusieurs emplacements. Les ressources ne sont pas répliquées à Régions AWS moins que vous ne le fassiez spécifiquement.

Amazon gère des centres de données à la pointe de la technologie et hautement disponibles. Bien qu'elles soient rares, des pannes touchant la disponibilité des instances se trouvant au même emplacement peuvent se produire. Si vous hébergez toutes vos instances dans un seul emplacement touché par une panne de ce type, aucune de vos instances ne sera disponible. Le schéma suivant montre une zone Région AWS comportant trois zones de disponibilité.



Il importe de se souvenir que chaque région est totalement indépendante. Toute activité Amazon DocumentDB que vous lancez (par exemple, la création d'instances ou la liste des instances disponibles) s'exécute uniquement dans votre configuration par défaut actuelle. Région AWS Vous pouvez également modifier la région par défaut dans la console, en définissant la variable d'environnement `EC2_REGION`. Ou vous pouvez la remplacer en utilisant le paramètre `--region` dans la AWS CLI. Pour plus d'informations, voir Configuration des AWS Command Line Interface sections spécifiques sur les variables d'environnement et les options de ligne de commande.

Lorsque vous créez un cluster à l'aide de la console Amazon DocumentDB et que vous choisissez de créer une réplique dans une autre zone de disponibilité, Amazon DocumentDB crée deux instances. Il crée l'instance principale dans une zone de disponibilité et l'instance de réplica dans une autre zone de disponibilité. Le volume de cluster est toujours répliqué sur trois zones de disponibilité.

Pour créer ou utiliser une instance Amazon DocumentDB dans une instance spécifique Région AWS, utilisez le point de terminaison de service régional correspondant.

Disponibilité dans les régions

Amazon DocumentDB est disponible dans les régions suivantes AWS .

Régions prises en charge par Amazon DocumentDB

Nom de la région	Région	Zones de disponibilité (calcul)
USA Est (Ohio)	`us-east-2`	3

Nom de la région	Région	Zones de disponibilité (calcul)
USA Est (Virginie du Nord)	us-east-1	6
USA Ouest (Oregon)	us-west-2	4
Amérique du Sud (São Paulo)	sa-east-1	3
Asie-Pacifique (Hong Kong)	ap-east-1	3
Asie-Pacifique (Hyderabad)	ap-south-2	3
Asie-Pacifique (Mumbai)	ap-south-1	3
Asie-Pacifique (Séoul)	ap-northeast-2	4
Asie-Pacifique (Singapour)	ap-southeast-1	3
Asie-Pacifique (Sydney)	ap-southeast-2	3
Asie-Pacifique (Tokyo)	ap-northeast-1	3
Canada (Centre)	ca-central-1	3
Région Chine (Beijing)	cn-north-1	3
Chine (Ningxia)	cn-northwest-1	3
Europe (Francfort)	eu-central-1	3
Europe (Irlande)	eu-west-1	3

Nom de la région	Région	Zones de disponibilité (calcul)
Europe (Londres)	`eu-west-2`	3
Europe (Milan)	`eu-south-1`	3
Europe (Paris)	`eu-west-3`	3
Moyen-Orient (EAU)	`me-central-1`	3
AWS GovCloud (US-Ouest)	`us-gov-west-1`	3
AWS GovCloud (USA Est)	`us-gov-east-1`	3

Par défaut, le fuseau horaire d'un cluster Amazon DocumentDB est Universal Time Coordinated (UTC).

Pour plus d'informations sur la recherche des points de terminaison de connexion pour les instances et les clusters dans une région donnée, consultez Comprendre les points de terminaison Amazon DocumentDB.

Gestion des groupes de paramètres du cluster Amazon DocumentDB

Vous pouvez gérer la configuration du moteur Amazon DocumentDB en utilisant les paramètres d'un groupe de paramètres de cluster. Un groupe de paramètres de cluster est un ensemble de valeurs de configuration Amazon DocumentDB qui facilitent la gestion des paramètres de vos clusters Amazon DocumentDB. Les groupes de paramètres de cluster servent de conteneurs pour les valeurs de configuration de moteur qui sont appliquées à toutes les instances du cluster.

Cette section décrit comment créer, afficher et modifier des groupes de paramètres de cluster. Elle explique également comment déterminer quel groupe de paramètres de cluster est associé à un cluster donné.

Rubriques

- Décrire les groupes de paramètres du cluster Amazon DocumentDB

- Création de groupes de paramètres de cluster Amazon DocumentDB
- Modification des groupes de paramètres du cluster Amazon DocumentDB
- Modification de clusters Amazon DocumentDB pour utiliser des groupes de paramètres de cluster personnalisés
- Copie des groupes de paramètres du cluster Amazon DocumentDB
- Réinitialisation des groupes de paramètres du cluster Amazon DocumentDB
- Suppression de groupes de paramètres de cluster Amazon DocumentDB
- Référence des paramètres du cluster Amazon DocumentDB

Décrire les groupes de paramètres du cluster Amazon DocumentDB

Un groupe de paramètres de `default` cluster est créé automatiquement lorsque vous créez le premier cluster Amazon DocumentDB dans une nouvelle région ou lorsque vous utilisez un nouveau moteur. Les clusters suivants, créés dans la même région et dotés de la même version de moteur, sont créés avec le groupe de paramètres de `default` cluster.

Rubriques

- Décrire les détails d'un groupe de paramètres de cluster Amazon DocumentDB
- Déterminer le groupe de paramètres d'un cluster Amazon DocumentDB

Décrire les détails d'un groupe de paramètres de cluster Amazon DocumentDB

Pour décrire les détails d'un groupe de paramètres de cluster donné, procédez comme suit à l'aide de l'AWS Management Console ou de l'AWS Command Line Interface (AWS CLI).

Using the AWS Management Console

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Groupes de paramètres.

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu

(≡)
dans le coin supérieur gauche de la page.

3. Dans le panneau Groupes de paramètres de cluster, sélectionnez le nom du groupe de paramètres de cluster dont vous voulez voir les détails.
4. La page qui s'affiche répertorie les paramètres du groupe de paramètres, l'activité récente ainsi que les balises.

   - Sous Cluster parameters (Paramètres de cluster), vous pouvez voir le nom du paramètre, la valeur actuelle, les valeurs autorisées, si le paramètre est modifiable, son type d'application, son type de données et sa description. Vous pouvez modifier des paramètres individuels en sélectionnant le paramètre, puis en choisissant Modifier dans la section Cluster parameters (Paramètres de cluster) . Pour plus d'informations, consultez Modification des paramètres du cluster Amazon DocumentDB.
   - Sous Événements récents, vous pouvez voir les événements les plus récents pour ce groupe de paramètres. Vous pouvez filtrer ces événements à l'aide de la barre de recherche de cette section. Pour plus d'informations, consultez Gestion des événements Amazon DocumentDB mentents.
   - Dans Tags (Balises), vous pouvez voir les balises qui se trouvent dans ce groupe de paramètres de cluster. Vous pouvez ajouter ou supprimer des balises en choisissant Modifier dans la section Balises. Pour plus d'informations, consultez Balisage des ressources Amazon DocumentDB.

Using the AWS CLI

Vous pouvez utiliser la `describe-db-cluster-parameter-groups` AWS CLI commande pour afficher le nom de ressource Amazon (ARN), la famille, la description et le nom d'un seul groupe de paramètres de cluster ou de tous les groupes de paramètres de cluster dont vous disposez pour Amazon DocumentDB. Vous pouvez également utiliser la commande `describe-db-cluster-parameters` de l'AWS CLI pour afficher les paramètres et leurs détails dans un seul groupe de paramètres de cluster.

- **`--describe-db-cluster-parameter-groups`**— Pour consulter la liste de tous vos groupes de paramètres de cluster et leurs détails.
  - **`--db-cluster-parameter-group-name`**— Facultatif. Nom du groupe de paramètres de cluster que vous voulez décrire. Si ce paramètre n'est pas spécifié, tous les groupes de paramètres de cluster sont décrits.

- **`--describe-db-cluster-parameters`**— Pour répertorier tous les paramètres d'un groupe de paramètres et leurs valeurs.
  - **`--db-cluster-parameter-group name`** — Obligatoire. Nom du groupe de paramètres de cluster que vous voulez décrire.

Example

Le code suivant répertorie jusqu'à 100 groupes de paramètres de cluster ainsi que leur ARN, leur famille, leur description et leur nom.

```
aws docdb describe-db-cluster-parameter-groups
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
          "DBClusterParameterGroups": [
              {
                  "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:012345678912:cluster-pg:default.docdb4.0",
                  "DBParameterGroupFamily": "docdb4.0",
                  "Description": "Default cluster parameter group for docdb4.0",
                  "DBClusterParameterGroupName": "default.docdb4.0"
              },
              {
                  "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:012345678912:cluster-pg:sample-parameter-group",
                  "DBParameterGroupFamily": "docdb4.0",
                  "Description": "Custom docdb4.0 parameter group",
                  "DBClusterParameterGroupName": "sample-parameter-group"
              }
          ]
}
```

Example

Le code suivant répertorie l'ARN, la famille, la description et le nom de `sample-parameter-group`.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-cluster-parameter-groups \
```

```
          --db-cluster-parameter-group-name sample-parameter-group
```

Pour Windows :

```
aws docdb describe-db-cluster-parameter-groups ^
          --db-cluster-parameter-group-name sample-parameter-group
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
          "DBClusterParameterGroups": [
              {
                  "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:sample-parameter-group",
                  "Description": "Custom docdb4.0 parameter group",
                  "DBParameterGroupFamily": "docdb4.0",
                  "DBClusterParameterGroupName": "sample-parameter-group"
              }
          ]
}
```

Example

Le code suivant répertorie les valeurs des paramètres dans `sample-parameter-group`.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-cluster-parameters \
    --db-cluster-parameter-group-name sample-parameter-group
```

Pour Windows :

```
aws docdb describe-db-cluster-parameters ^
    --db-cluster-parameter-group-name sample-parameter-group
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
   "Parameters": [
         {
            "ParameterName": "audit_logs",
            "ParameterValue": "disabled",
            "Description": "Enables auditing on cluster.",
            "Source": "system",
            "ApplyType": "dynamic",
            "DataType": "string",
            "AllowedValues": "enabled,disabled",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot"
         },
         {
            "ParameterName": "change_stream_log_retention_duration",
            "ParameterValue": "17777",
            "Description": "Duration of time in seconds that the change stream log
 is retained and can be consumed.",
            "Source": "user",
            "ApplyType": "dynamic",
            "DataType": "integer",
            "AllowedValues": "3600-86400",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot"
         }
   ]
}
```

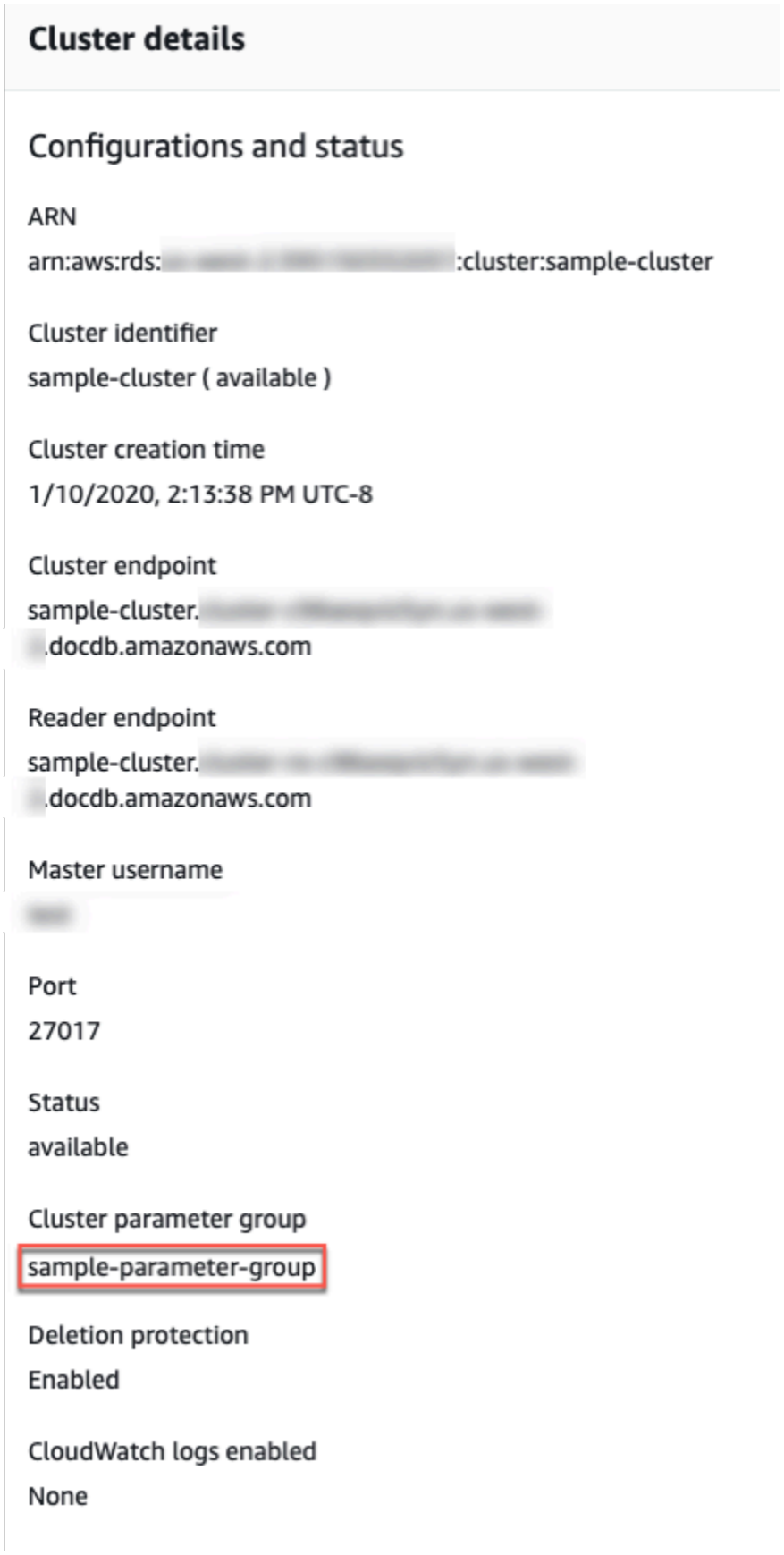
Déterminer le groupe de paramètres d'un cluster Amazon DocumentDB

Pour déterminer quel groupe de paramètres est associé à un cluster particulier, procédez comme suit à l'aide de l'AWS Management Console ou de l'AWS CLI.

Using the AWS Management Console

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation de gauche, choisissez Clusters.
3. Dans la liste des clusters, sélectionnez le nom du cluster qui vous intéresse.

4. La page qui s'affiche répertorie les détails du cluster sélectionné. Faites défiler vers le bas jusqu'à Cluster details (Détails du cluster). En bas de cette section, recherchez le nom du groupe de paramètres sous Cluster parameter group (Groupe de paramètres de cluster).


Cluster details
Configurations and status
ARN
arn:aws:rds:[redacted]:cluster:sample-cluster
Cluster identifier
sample-cluster (available)
Cluster creation time
1/10/2020, 2:13:38 PM UTC-8
Cluster endpoint
sample-cluster.[redacted]
.docdb.amazonaws.com
Reader endpoint
sample-cluster.[redacted]
.docdb.amazonaws.com
Master username
[redacted]
Port
27017
Status
available
Cluster parameter group
sample-parameter-group
Deletion protection
Enabled
CloudWatch logs enabled
None

Using the AWS CLI

Le code suivant de l'AWS CLI détermine quel groupe de paramètres régit le `sample-cluster` du cluster.

```
aws docdb describe-db-clusters \
      --db-cluster-identifier sample-cluster \
      --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
```

La sortie de cette opération ressemble à ceci (format JSON).

```
[
       [
           "sample-cluster",
           "sample-parameter-group"
       ]
]
```

Création de groupes de paramètres de cluster Amazon DocumentDB

Les groupes de paramètres de cluster par défaut tels que `default.docdb5.0` `default.docdb4.0default.docdb3.6`, ou sont créés lorsque vous créez un cluster avec une nouvelle version du moteur et dans une nouvelle région. Les clusters suivants créés dans cette région et dotés de la même version de moteur héritent du groupe de paramètres de `default` cluster. Une fois créés, les groupes de `default` paramètres ne peuvent pas être supprimés ou renommés. Vous pouvez modifier le comportement du moteur des instances de cluster en créant un groupe de paramètres personnalisé avec des valeurs de paramètres préférées et en l'attachant à votre cluster Amazon DocumentDB.

La procédure suivante vous guide tout au long de la création d'un groupe de paramètres de cluster personnalisé. Vous pouvez ensuite modifier les paramètres de ce groupe de paramètres.

> **Note**
>
> Après la création d'un groupe de paramètres de cluster, patientez au moins 5 minutes avant d'utiliser ce groupe de paramètres de cluster. Cela permet à Amazon DocumentDB de terminer complètement l'`create`action avant que le groupe de paramètres du cluster ne soit utilisé pour un nouveau cluster. Vous pouvez utiliser l'AWS Management Console ou l'opération `describe-db-cluster-parameter-groups` de l'AWS CLI pour vérifier que

votre groupe de paramètres de cluster a été créé. Pour plus d'informations, consultez Décrire les groupes de paramètres du cluster Amazon DocumentDB.

Using the AWS Management Console

Créer un groupe de paramètres de cluster

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Groupes de paramètres.

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans le panneau Groupes de paramètres de cluster, choisissez Créer.
4. Dans le panneau Create cluster parameter group (Créer un groupe de paramètres de cluster) entrez les éléments suivants :

   a. Nom du groupe — Entrez le nom du groupe de paramètres du cluster. Par exemple, `sample-parameter-group`. Les groupes de paramètres de cluster sont soumis aux contraintes de dénomination suivantes :

      - Entre 1 et 255 caractères alphanumériques.
      - Le premier caractère doit être une lettre.
      - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

   b. Description — Fournissez une description de ce groupe de paramètres de cluster.
5. Choisissez Créer pour créer le groupe de paramètres de cluster. Pour annuler l'opération, choisissez Annuler.
6. Une fois que vous avez sélectionné Créer, le texte suivant s'affiche en haut de la page pour confirmer que votre groupe de paramètres de cluster a été créé avec succès :

```
Successfully created cluster parameter group 'sample-parameter-group'.
```

Using the AWS CLI

Pour créer un nouveau groupe de paramètres de cluster pour les clusters Amazon DocumentDB 4.0, utilisez l'AWS CLI`create-db-cluster-parameter-group`opération avec les paramètres suivants :

- **`--db-cluster-parameter-group-name`**— Nom du groupe de paramètres de cluster personnalisé. Par exemple, `sample-parameter-group`.
- **`--db-cluster-parameter-group-family`**— La famille de groupes de paramètres de cluster utilisée comme modèle pour le groupe de paramètres de cluster personnalisé. Actuellement, il doit s'agir de `docdb4.0`.
- **`--description`**— Description fournie par l'utilisateur pour ce groupe de paramètres de cluster. L'exemple suivant utilise `Custom docdb4.0 parameter group`.

Pour Linux, macOS ou Unix :

Example

```
aws docdb create-db-cluster-parameter-group \
 --db-cluster-parameter-group-name sample-parameter-group \
 --db-parameter-group-family docdb4.0 \
 --description "Custom docdb4.0 parameter group"
```

Pour Windows :

```
aws docdb create-db-cluster-parameter-group ^
 --db-cluster-parameter-group-name sample-parameter-group ^
 --db-parameter-group-family docdb4.0 ^
 --description "Custom docdb4.0 parameter group"
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "DBClusterParameterGroup": {
        "DBClusterParameterGroupName": "sample-parameter-group",
        "DBParameterGroupFamily": "docdb4.0",
        "Description": "Custom docdb4.0 parameter group",
        "DBClusterParameterGroupArn": "sample-parameter-group-arn"
    }
```

```
}
```

Modification des groupes de paramètres du cluster Amazon DocumentDB

Cette section explique comment modifier un groupe de paramètres Amazon DocumentDB personnalisé. Dans Amazon DocumentDB, vous ne pouvez pas modifier un groupe de paramètres de `default` cluster créé lorsque vous créez pour la première fois un cluster avec une nouvelle version du moteur dans une nouvelle région. Si votre cluster Amazon DocumentDB utilise le groupe de paramètres de cluster par défaut et que vous souhaitez y modifier une valeur, vous devez d'abord créer un nouveau groupe de paramètres ou copier un groupe de paramètres existant, le modifier, puis appliquer le groupe de paramètres modifié à votre cluster.

Procédez comme suit pour modifier un groupe de paramètres de cluster personnalisé. La propagation des actions de modification peut prendre un certain temps. Attendez que le groupe de paramètres de cluster modifié soit disponible avant de l'associer à votre cluster. Vous pouvez utiliser l'AWS Management Console ou l'opération `describe-db-cluster-parameters` de l'AWS CLI pour vérifier que votre groupe de paramètres de cluster a été modifié. Pour plus d'informations, consultez Décrire les groupes de paramètres de cluster.

Using the AWS Management Console

Suivez ces étapes pour modifier un groupe de paramètres Amazon DocumentDB personnalisé. Vous ne pouvez pas modifier un groupe de paramètres `default`. Si vous souhaitez modifier une valeur dans le groupe de paramètres `default`, vous pouvez copier le groupe de paramètres de cluster par défaut, le modifier, puis l'appliquer à votre cluster. Pour de plus amples informations sur l'application de groupes de paramètres à votre cluster, veuillez consulter Modification d'un cluster Amazon DocumentDB.

Pour modifier un groupe de paramètres de cluster personnalisé

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation sur le côté gauche de la console, choisissez Groupes de paramètres. Dans la liste des groupes de paramètres, sélectionnez le nom du groupe de paramètres que vous souhaitez modifier.

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Pour chaque paramètre dans le groupe de paramètres que vous souhaitez modifier, procédez comme suit :

   a. Recherchez le paramètre que vous souhaitez modifier et vérifiez qu'il est modifiable : `true` doit être indiqué dans la colonne Modifiable .

   b. S'il est modifiable, sélectionnez le paramètre et choisissez Modifier dans l'angle supérieur droit de la page de la console.

   c. Dans la boîte `<parameter-name>` de dialogue Modifier, apportez les modifications souhaitées. Choisissez ensuite Modify cluster parameter (Modifier le paramètre de cluster), ou Annuler pour annuler les modifications.

Using the AWS CLI

Vous pouvez modifier le `ParameterValue` ou `ApplyMethod` de n'importe quel paramètre modifiable dans un groupe de paramètres de cluster Amazon DocumentDB personnalisé à l'aide du. `Description` AWS CLI Vous ne pouvez pas apporter de modifications directement à un groupe de paramètres de cluster par défaut.

Pour modifier les paramètres d'un groupe de paramètres de cluster personnalisé, utilisez l'opération `modify-db-cluster-parameter-group` avec les paramètres suivants.

- **`--db-cluster-parameter-group-name`** — Obligatoire. Nom du groupe de paramètres de cluster que vous voulez modifier.
- **`--parameters`** — Obligatoire. Paramètres que vous modifiez. Pour obtenir la liste des paramètres qui s'appliquent à toutes les instances d'un cluster Amazon DocumentDB, consultez le. Référence des paramètres du cluster Amazon DocumentDB Chaque saisie de paramètre doit inclure ce qui suit :
  - **`ParameterName`**— Le nom du paramètre que vous êtes en train de modifier.
  - **`ParameterValue`**— La nouvelle valeur de ce paramètre.

- **`ApplyMethod`**— La manière dont vous souhaitez que les modifications soient appliquées à ce paramètre. Les valeurs autorisées sont `immediate` et `pending-reboot`.

> **Note**
>
> Les paramètres avec le `ApplyType` de `static` doivent avoir une `ApplyMethod` de `pending-reboot`.

Example - Modification de la valeur d'un paramètre

Dans cet exemple, vous répertoriez les valeurs du paramètre du `sample-parameter-group` et modifiez le paramètre `tls`. Ensuite, après 5 minutes, vous répertoriez à nouveau les valeurs du `sample-parameter-group` pour voir les valeurs du paramètre modifiées.

1. Répertoriez les paramètres et leurs valeurs de `sample-parameter-group`.

   Pour Linux, macOS ou Unix :

   ```
   aws docdb describe-db-cluster-parameters \
          --db-cluster-parameter-group-name sample-parameter-group
   ```

   Pour Windows :

   ```
   aws docdb describe-db-cluster-parameters ^
          --db-cluster-parameter-group-name sample-parameter-group
   ```

   La sortie de cette opération ressemble à ceci (format JSON).

   ```
   {
          "Parameters": [
              {
                  "Source": "system",
                  "ApplyType": "static",
                  "AllowedValues": "disabled,enabled",
                  "ParameterValue": "enabled",
                  "ApplyMethod": "pending-reboot",
                  "DataType": "string",
                  "ParameterName": "tls",
                  "IsModifiable": true,
   ```

```
                "Description": "Config to enable/disable TLS"
            },
            {
                "Source": "user",
                "ApplyType": "dynamic",
                "AllowedValues": "disabled,enabled",
                "ParameterValue": "enabled",
                "ApplyMethod": "pending-reboot",
                "DataType": "string",
                "ParameterName": "ttl_monitor",
                "IsModifiable": true,
                "Description": "Enables TTL Monitoring"
            }
        ]
}
```

2. Modifiez le paramètre `tls` afin que sa valeur soit `disabled`.

   Vous ne pouvez pas modifier la `ApplyMethod` car le `ApplyType` est `static`.

   Pour Linux, macOS ou Unix :

   ```
   aws docdb modify-db-cluster-parameter-group \
         --db-cluster-parameter-group-name sample-parameter-group \
         --parameters
    "ParameterName"=tls,"ParameterValue"=disabled,"ApplyMethod"=pending-reboot
   ```

   Pour Windows :

   ```
   aws docdb modify-db-cluster-parameter-group ^
         --db-cluster-parameter-group-name sample-parameter-group ^
         --parameters
    "ParameterName"=tls,"ParameterValue"=disabled,"ApplyMethod"=pending-reboot
   ```

   La sortie de cette opération ressemble à ceci (format JSON).

   ```
   {
         "DBClusterParameterGroupName": "sample-parameter-group"
     }
   ```

3. Patientez au moins 5 minutes.

4. Répertoriez les valeurs des paramètres de `sample-parameter-group` pour vérifier que le paramètre `tls` a été modifié.

   Pour Linux, macOS ou Unix :

```
aws docdb describe-db-cluster-parameters \
       --db-cluster-parameter-group-name sample-parameter-group
```

   Pour Windows :

```
aws docdb describe-db-cluster-parameters ^
       --db-cluster-parameter-group-name sample-parameter-group
```

   La sortie de cette opération ressemble à ceci (format JSON).

```
{
       "Parameters": [
           {
               "ParameterValue": "false",
               "ParameterName": "enable_audit_logs",
               "ApplyType": "dynamic",
               "DataType": "string",
               "Description": "Enables auditing on cluster.",
               "AllowedValues": "true,false",
               "Source": "system",
               "IsModifiable": true,
               "ApplyMethod": "pending-reboot"
           },
           {
               "ParameterValue": "disabled",
               "ParameterName": "tls",
               "ApplyType": "static",
               "DataType": "string",
               "Description": "Config to enable/disable TLS",
               "AllowedValues": "disabled,enabled",
               "Source": "system",
               "IsModifiable": true,
               "ApplyMethod": "pending-reboot"
           }
       ]
}
```

Modification de clusters Amazon DocumentDB pour utiliser des groupes de paramètres de cluster personnalisés

Lorsque vous créez un cluster Amazon DocumentDB, un groupe de `default.docdb4.0` paramètres est automatiquement créé pour ce cluster. Vous ne pouvez pas modifier le groupe de paramètres de cluster `default`. Au lieu de cela, vous pouvez modifier votre cluster Amazon DocumentDB pour y associer un nouveau groupe de paramètres personnalisés.

Cette section explique comment modifier un cluster Amazon DocumentDB existant pour utiliser un groupe de paramètres de cluster personnalisé à l'aide du AWS Management Console et AWS Command Line Interface ()AWS CLI.

Using the AWS Management Console

Pour modifier un cluster Amazon DocumentDB afin d'utiliser un nouveau groupe de paramètres de cluster autre que celui par défaut

1. Avant de commencer, assurez-vous d'avoir créé un cluster Amazon DocumentDB et un groupe de paramètres de cluster. Pour consulter des instructions supplémentaires, veuillez consulter Création d'un cluster Amazon DocumentDB et Création de groupes de paramètres de cluster Amazon DocumentDB.
2. Après avoir créé votre groupe de paramètres de cluster, ouvrez la console Amazon DocumentDB à l'adresse. https://console.aws.amazon.com/docdb Dans le panneau de navigation, choisissez Clusters pour ajouter votre nouveau groupe de paramètres à un cluster.
3. Choisissez le cluster auquel vous souhaitez associer votre groupe de paramètres. Choisissez Actions, puis Modifier pour modifier votre cluster.
4. Sous Cluster options (Options de cluster), choisissez le nouveau groupe de paramètres auquel vous souhaitez associer votre cluster.
5. Sélectionnez Continuer pour afficher un résumé de vos modifications.
6. Après vérification de vos modifications, vous pouvez les appliquer immédiatement ou au cours de la fenêtre de maintenance suivante sous Scheduling of modifications (Planification des modifications).
7. Choisissez Modify cluster (Modifier le cluster) pour mettre à jour votre cluster avec votre nouveau groupe de paramètres.

Using the AWS CLI

Avant de commencer, assurez-vous d'avoir créé un cluster Amazon DocumentDB et un groupe de paramètres de cluster. Vous pouvez créer un cluster Amazon DocumentDB à l'aide de cette opération. AWS CLI `create-db-cluster` Vous pouvez créer un groupe de paramètres de cluster à l'aide de l'opération `create-db-cluster-parameter-group` de l'AWS CLI.

Pour ajouter votre nouveau groupe de paramètres de cluster à votre cluster, utilisez l'opération `modify-db-cluster` de l'AWS CLI avec les paramètres suivants.

- -- db-cluster-identifier — Le nom de votre cluster (par exemple,`sample-cluster`).
- -- db-cluster-parameter-group -name — Le nom du groupe de paramètres auquel vous souhaitez associer votre cluster (par exemple,`sample-parameter-group`).

Example

```
aws docdb modify-db-cluster \
    --db-cluster-identifier sample-cluster
    --db-cluster-parameter-group-name sample-parameter-group
```

La sortie de cette opération ressemble à ceci (format JSON).

```
"DBCluster": {
    "AvailabilityZones": [
            "us-west-2c",
            "us-west-2b",
            "us-west-2a"
    ],
    "BackupRetentionPeriod": 1,
    "DBClusterIdentifier": "sample-cluster",
    "DBClusterParameterGroup": "sample-parameter-group",
    "DBSubnetGroup": "default",
   ...
}
```

Copie des groupes de paramètres du cluster Amazon DocumentDB

Vous pouvez créer une copie d'un groupe de paramètres de cluster dans Amazon DocumentDB à l'aide du AWS Management Console ou du AWS Command Line Interface ()AWS CLI.

Using the AWS Management Console

La procédure suivante vous guide tout au long de la création d'un nouveau groupe de paramètres de cluster en effectuant une copie d'un groupe de paramètres de cluster existant.

Pour copier un groupe de paramètres de cluster

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Groupes de paramètres.
3. Dans le panneau Groupes de paramètres de cluster, sélectionnez le nom du groupe de paramètres de cluster que vous souhaitez copier.
4. Choisissez Actions, puis Copy (Copier) pour copier ce groupe de paramètres.
5. Sous Copy options (Options de copie), entrez le nom et une description du nouveau groupe de paramètres de cluster. Choisissez ensuite Copy (Copier) pour enregistrer vos modifications.

Using the AWS CLI

Pour réaliser une copie d'un groupe de paramètres de cluster, utilisez l'opération `copy-db-cluster-parameter-group` avec les paramètres suivants.

- **`--source-db-cluster-parameter-group-identifier`** — Obligatoire. Nom ou Amazon Resource Name (ARN) du groupe de paramètres de cluster dont vous voulez réaliser une copie.

  Si les groupes de paramètres du cluster source et cible sont identiquesRégion AWS, l'identifiant peut être un nom ou un ARN.

  Si les groupes de paramètres du cluster source et cible sont différentsRégions AWS, l'identifiant doit être un ARN.
- **`--target-db-cluster-parameter-group-identifier`** — Obligatoire. Le nom ou l'ARN de la copie du groupe de paramètres du cluster.

  Contraintes :

  - Ne peut pas être null ou vide.
  - Doit contenir de 1 à 255 lettres, chiffres ou traits d'union.

  - Le premier caractère doit être une lettre.
  - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.
- **`--target-db-cluster-parameter-group-description`** — Obligatoire. Un utilisateur a fourni une description pour la copie du groupe de paramètres de cluster.

Example

Le code suivant effectue une copie de `sample-parameter-group`, en appelant la copie `sample-parameter-group-copy`.

Pour Linux, macOS ou Unix :

```
aws docdb copy-db-cluster-parameter-group \
    --source-db-cluster-parameter-group-identifier sample-parameter-group \
    --target-db-cluster-parameter-group-identifier sample-parameter-group-copy \
    --target-db-cluster-parameter-group-description "Copy of sample-parameter-group"
```

Pour Windows :

```
aws docdb copy-db-cluster-parameter-group ^
    --source-db-cluster-parameter-group-identifier sample-parameter-group ^
    --target-db-cluster-parameter-group-identifier sample-parameter-group-copy ^
    --target-db-cluster-parameter-group-description "Copy of sample-parameter-group"
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "DBClusterParameterGroup": {
        "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:sample-parameter-group-copy",
        "DBClusterParameterGroupName": "sample-parameter-group-copy",
        "DBParameterGroupFamily": "docdb4.0",
        "Description": "Copy of sample-parameter-group"
    }
}
```

Réinitialisation des groupes de paramètres du cluster Amazon DocumentDB

Vous pouvez rétablir certaines ou toutes les valeurs des paramètres d'un groupe de paramètres de cluster Amazon DocumentDB à leurs valeurs par défaut en utilisant le AWS Management Console ou le AWS Command Line Interface (AWS CLI) pour réinitialiser le groupe de paramètres de cluster.

Using the AWS Management Console

Procédez comme suit pour réinitialiser certaines ou toutes les valeurs de paramètres d'un groupe de paramètres de cluster à leurs valeurs par défaut.

Pour réinitialiser les valeurs de paramètres d'un groupe de paramètres de cluster

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation sur le côté gauche de la console, choisissez Groupes de paramètres.
3. Dans le panneau Groupes de paramètres de cluster, choisissez le nom du groupe de paramètres de cluster que vous souhaitez réinitialiser.
4. Choisissez Actions, puis Réinitialiser pour réinitialiser ce groupe de paramètres.
5. Sur la page Cluster parameter group reset confirmation (Confirmation de réinitialisation du groupe de paramètres de cluster) qui s'affiche, confirmez que vous souhaitez réinitialiser tous les paramètres de cluster de ce groupe de paramètres à leurs valeurs par défaut. Choisissez ensuite Réinitialiser pour réinitialiser votre groupe de paramètres. Vous pouvez également choisir Annuler pour annuler vos modifications.

Using the AWS CLI

Pour réinitialiser certaines ou toutes les valeurs des paramètres d'un groupe de paramètres de cluster aux valeurs par défaut, utilisez l'opération `reset-db-cluster-parameter-group` avec les paramètres suivants.

- **`--db-cluster-parameter-group-name`** — Obligatoire. Le nom du groupe de paramètres du cluster à réinitialiser.
- **`--parameters`**— Facultatif. Une liste des `ParameterName` et de la `ApplyMethod` dans le groupe de paramètres de cluster à réinitialiser à leurs valeurs par défaut. Les paramètres statiques doit être définie sur `pending-reboot` pour entrer en vigueur lors du prochain

redémarrage de l'instance ou de la demande `reboot-db-instance`. Vous devez appeler `reboot-db-instance` pour chaque instance de votre cluster à laquelle vous souhaitez que la mise à jour des paramètres statiques soit appliquée.

Ce paramètre et `--reset-all-parameters` s'excluent mutuellement : vous pouvez utiliser l'un ou l'autre, mais pas les deux.

- **`--reset-all-parameters`**ou **`--no-reset-all-parameters`** — Facultatif. Spécifie s'il faut réinitialiser tous les paramètres (`--reset-all-parameters`) ou uniquement certains des paramètres (`--no-reset-all-parameters`) à leurs valeurs par défaut. Le paramètre `--reset-all-parameters` et `--parameters` s'excluent mutuellement : vous pouvez utiliser l'un ou l'autre, mais pas les deux.

  Lorsque vous réinitialisez l'ensemble du groupe, les paramètres dynamiques sont mis à jour immédiatement. Les paramètres statiques doivent être définis sur `pending-reboot` pour entrer en vigueur lors du redémarrage suivant de l'instance ou de la demande `reboot-db-instance`. Vous devez appeler `reboot-db-instance` pour chaque instance de votre cluster à laquelle vous souhaitez que la mise à jour des paramètres statiques soit appliquée.

Example

Exemple 1 : Réinitialisation de tous les paramètres à leurs valeurs par défaut

Le code suivant réinitialise tous les paramètres dans le groupe de paramètres de cluster `sample-parameter-group` à leurs valeurs par défaut.

Pour Linux, macOS ou Unix :

```
aws docdb reset-db-cluster-parameter-group \
       --db-cluster-parameter-group-name sample-parameter-group \
       --reset-all-parameters
```

Pour Windows :

```
aws docdb reset-db-cluster-parameter-group ^
       --db-cluster-parameter-group-name sample-parameter-group ^
       --reset-all-parameters
```

Exemple 2 : Réinitialisation de paramètres spécifiés à leurs valeurs par défaut

Le code suivant réinitialise le paramètre `tls` dans le groupe de paramètres de cluster `sample-parameter-group` à sa valeur par défaut.

Pour Linux, macOS ou Unix :

```
aws docdb reset-db-cluster-parameter-group \
       --db-cluster-parameter-group-name sample-parameter-group \
       --no-reset-all-parameters \
       --parameters ParameterName=tls,ApplyMethod=pending-reboot
```

Pour Windows :

```
aws docdb reset-db-cluster-parameter-group ^
       --db-cluster-parameter-group-name sample-parameter-group ^
       --no-reset-all-parameters ^
       --parameters ParameterName=tls,ApplyMethod=pending-reboot
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
       "DBClusterParameterGroupName": "sample-parameter-group"
   }
```

Redémarrage d'une instance de cluster

Avant qu'une valeur de paramètre statique soit modifiée, l'instance du cluster doit être redémarrée. Redémarrez chaque instance de votre cluster à laquelle vous souhaitez que la mise à jour du paramètre statique soit appliquée.

Pour Linux, macOS ou Unix :

```
aws docdb reboot-db-instance \
       --db-instance-identifier sample-cluster-instance
```

Pour Windows :

```
aws docdb reboot-db-instance ^
       --db-instance-identifier sample-cluster-instance
```

Suppression de groupes de paramètres de cluster Amazon DocumentDB

Vous pouvez supprimer un groupe de paramètres de cluster Amazon DocumentDB personnalisé à l'aide du AWS Management Console ou du AWS Command Line Interface ()AWS CLI. Vous ne pouvez pas supprimer le groupe de paramètres de cluster `default.docdb4.0`.

Using the AWS Management Console

Supprimer un groupe de paramètres de cluster

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Groupes de paramètres.

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans le volet Groupes de paramètres, sélectionnez la case d'option à gauche du groupe de paramètres de cluster que vous voulez supprimer.
4. Choisissez Actions, puis Delete (Supprimer).
5. Dans le volet de confirmation Supprimer choisissez Supprimer pour supprimer le groupe de paramètres de cluster. Pour conserver le groupe de paramètres de cluster, choisissez Annuler.

Using the AWS CLI

Pour supprimer un groupe de paramètres de cluster, utilisez l'opération `delete-db-cluster-parameter-group` avec le paramètre suivant.

- **`--db-cluster-parameter-group-name`** — Obligatoire. Le nom du groupe de paramètres du cluster à supprimer. Il doit s'agir d'un groupe de paramètres de cluster existant. Vous ne pouvez pas supprimer le groupe de paramètres de cluster *`default.docdb4.0`*.

Example - Suppression d'un groupe de paramètres de cluster

Suivez les trois étapes de l'exemple suivant pour la suppression d'un groupe de paramètres de cluster :

1. Trouver le nom du groupe de paramètres de cluster que vous voulez supprimer.
2. Suppression du groupe de paramètres de cluster spécifié.
3. Vérification de la suppression du groupe de paramètres de cluster.

1. Trouvez le nom du groupe de paramètres de cluster que vous voulez supprimer.

Le code suivant répertorie les noms de tous les groupes de paramètres de cluster.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-cluster-parameter-groups \
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Pour Windows :

```
aws docdb describe-db-cluster-parameter-groups ^
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Le résultat de l'opération précédente est une liste des noms des groupes de paramètres de cluster similaire à celle ci-après (au format JSON).

```
[
       [
           "default.docdb4.0"
       ],
       [
           "sample-parameter-group"
       ],
       [
           "sample-parameter-group-copy"
       ]
   ]
```

2. Supprimer un groupe de paramètres de cluster spécifique.

Le code suivant supprime le groupe de paramètres de cluster `sample-parameter-group-copy`.

Pour Linux, macOS ou Unix :

```
aws docdb delete-db-cluster-parameter-group \
       --db-cluster-parameter-group-name sample-parameter-group-copy
```

Pour Windows :

```
aws docdb delete-db-cluster-parameter-group ^
       --db-cluster-parameter-group-name sample-parameter-group-copy
```

Il n'existe aucun résultat pour cette opération.

3. Vérifiez que le groupe de paramètres de cluster a été supprimé.

Le code suivant répertorie les noms de tous les groupes de paramètres de cluster restants.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-cluster-parameter-groups \
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Pour Windows :

```
aws docdb describe-db-cluster-parameter-groups ^
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Le résultat de l'opération précédente est une liste des groupes de paramètres de cluster similaire à celle ci-après (au format JSON). Le groupe de paramètres de cluster que vous venez de supprimer ne doit pas figurer dans la liste.

La sortie de cette opération ressemble à ceci (format JSON).

```
[
       [
           "default.docdb4.0"
       ],
       [
```

```
            "sample-parameter-group"
        ]
    ]
```

Référence des paramètres du cluster Amazon DocumentDB

Lorsque vous modifiez un paramètre dynamique et que vous enregistrez le groupe de paramètres de cluster, la modification est appliquée immédiatement, quel que soit le paramètre Apply Immediately (Appliquer immédiatement). Lorsque vous modifiez un paramètre statique et que vous enregistrez le groupe de paramètres de cluster, la modification du paramètre est appliquée après que vous avez redémarré manuellement l'instance. Vous pouvez redémarrer une instance à l'aide de la console Amazon DocumentDB ou en appelant explicitement. `reboot-db-instance`

Le tableau suivant indique les paramètres qui s'appliquent à toutes les instances d'un cluster Amazon DocumentDB.

Paramètres au niveau du cluster Amazon DocumentDB

Paramètre	Valeur par défaut	Valeurs valides	Adaptabilité	Type d'application	Type de données	Description
`audit_logs`	disabled	activé, désactivé, ddl, dml_read, dml_write, tout, aucun	Oui	Répartition dynamique	Chaîne	Définit si les journaux CloudWatch d'audit Amazon sont activés. • **`enabled`**— les journaux CloudWatch d'audit sont activés.

Paramètre	Valeur par défaut	Valeurs valides	Adaptabilité	Type d'application	Type de données	Description
						• **disabled**— les journaux CloudWatch d'audit sont désactivés. • **ddl**— l'audit des événements DDL est activé. • **dml_read**— l'audit des événements de lecture DML est activé. • **dml_write** — l'audit des événements d'écriture DML est activé.

Paramètre	Valeur par défaut	Valeurs valides	Adaptabilité	Type d'application	Type de données	Description
						• **all**— l'audit de tous les événements de base de données est activé. • **none**— l'audit est désactivé.
change_stream_log_retention_duration	10800	3600-604800	Oui	Répartition dynamique	Entier	Définit la durée (en secondes) pendant laquelle le journal du flux de modifications est conservé et peut être consommé.

Paramètre	Valeur par défaut	Valeurs valides	Adaptabilité	Type d'application	Type de données	Description
`profiler`	disabled	activé, désactivé	Oui	Répartition dynamique	Chaîne	Active le profilage pour les opérations lentes. • **`enabled`**— les opérations qui prennent plus de temps qu'une valeur seuil définie par le client (100 ms, par exemple) sont enregistrées dans Amazon Logs. CloudWatch • **`disabled`**— les opérations lentes

Paramètre	Valeur par défaut	Valeurs valides	Adaptabilité	Type d'application	Type de données	Description
						ne sont pas enregistrées dans les CloudWatch journaux.
`profiler_sampling_rate`	1.0	0.0-1.0	Oui	Répartition dynamique	Float	Définit la fréquence d'échantillonnage pour les opérations consignées.

Paramètre	Valeur par défaut	Valeurs valides	Adaptabilité	Type d'application	Type de données	Description
`profiler_threshold_ms`	100	50-2147483646	Oui	Répartition dynamique	Entier	Définit le seuil pour `profiler`. • Toutes les opérations supérieures à `profiler_threshold_ms` sont enregistrées dans CloudWatch Logs.

Paramètre	Valeur par défaut	Valeurs valides	Adaptabilité	Type d'application	Type de données	Description
`tls`	activé	activé, désactivé, fips-140-3	Oui	Statique	Chaîne	Définit si des connexions utilisant le protocole TLS sont requises. • **`enabled`**— Des connexions TLS sont nécessaires pour se connecter. • **`disabled`**— Les connexions TLS ne peuvent pas être utilisées pour se connecter. • **`fips-140-3`** —Des connexions TLS

Paramètre	Valeur par défaut	Valeurs valides	Adaptabilité	Type d'application	Type de données	Description
						avec des attributs FIPS (Federal Information Processing Standards) sont nécessaires pour se connecter. Le cluster accepte uniquement les connexions sécurisées conformément à la publication FIPS 140-3. Ceci n'est pris en charge qu'à

Paramètre	Valeur par défaut	Valeurs valides	Adaptabilité	Type d'application	Type de données	Description
						partir des clusters Amazon DocumentDB 5.0 (moteur version 3.0.3727) dans les régions suivantes : ca-central-1, us-west-2, us-east-1, us-east-2, -1, -1. us-gov-east us-gov-west

Paramètre	Valeur par défaut	Valeurs valides	Adaptabilité	Type d'application	Type de données	Description
`ttl_monitor`	activé	activé, désactivé	Oui	Répartition dynamique	Chaîne	Définit si la surveillance de durée de vie (TTL) est activée pour le cluster. • **`enabled`**— La surveillance TTL est activée. • **`disabled`**— La surveillance TTL est désactivée.

Modification des paramètres du cluster Amazon DocumentDB

Dans Amazon DocumentDB, les groupes de paramètres de cluster sont des paramètres qui s'appliquent à toutes les instances que vous créez dans le cluster. Pour les groupes de paramètres de cluster personnalisés, vous pouvez modifier une valeur de paramètre à tout moment ou réinitialiser toutes les valeurs de paramètre à leurs valeurs par défaut pour les groupes de paramètres que vous créez. Cette section explique comment afficher les paramètres qui constituent un groupe de paramètres de cluster Amazon DocumentDB et leurs valeurs, et comment modifier ou mettre à jour ces valeurs.

Les paramètres peuvent être dynamiques ou statiques. Lorsque vous modifiez un paramètre dynamique et que vous enregistrez le groupe de paramètres de cluster, la modification est appliquée immédiatement, quel que soit le paramètre `Apply Immediately`. Lorsque vous modifiez un paramètre statique et que vous enregistrez le groupe de paramètres de cluster, la modification du paramètre est appliquée après que vous avez redémarré manuellement l'instance.

Afficher les paramètres d'un groupe de paramètres de cluster Amazon DocumentDB

Vous pouvez consulter les paramètres d'un cluster Amazon DocumentDB et leurs valeurs à l'aide du AWS Management Console ou. AWS CLI

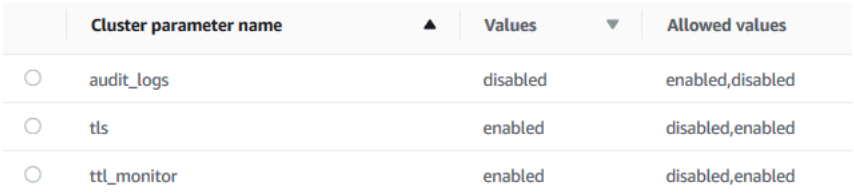
Using the AWS Management Console

Pour afficher les détails d'un groupe de paramètres de cluster

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Groupes de paramètres.

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans le volet Groupes de paramètres, sélectionnez le nom du groupe de paramètres de cluster dont vous voulez voir les détails.
4. La page qui s'affiche montre les valeurs suivantes pour chaque paramètre : nom du paramètre, valeur actuelle, valeurs autorisées, si le paramètre est modifiable, type d'application, type de données et description.



	Cluster parameter name ▲	Values ▼	Allowed values
○	audit_logs	disabled	enabled,disabled
○	tls	enabled	disabled,enabled
○	ttl_monitor	enabled	disabled,enabled

Using the AWS CLI

Pour afficher les paramètres et les valeurs d'un groupe de paramètres de cluster, utilisez l'opération `describe-db-cluster-parameters` avec les paramètres suivants.

- **`--db-cluster-parameter-group-name`** — Obligatoire. Le nom du groupe de paramètres de cluster pour lequel vous souhaitez obtenir une liste détaillée des paramètres.
- **`--source`**— Facultatif. S'il est fournit, il renvoie uniquement les paramètres pour une source spécifique. Les sources du paramètre peuvent être `engine-default`, `system` ou `user`.

Example

L'exemple de code suivant répertorie les paramètres et leurs valeurs pour le groupe de paramètres `custom3-6-param-grp`. Pour obtenir des informations sur le groupe de paramètres, omettez la ligne `--query`. Pour obtenir des informations sur tous les groupes de paramètres, omettez la ligne `--db-cluster-parameter-group-name`.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-cluster-parameters \
   --db-cluster-parameter-group-name custom3-6-param-grp \
   --query 'Parameters[*].[ParameterName,ParameterValue]'
```

Pour Windows :

```
aws docdb describe-db-cluster-parameters ^
   --db-cluster-parameter-group-name custom3-6-param-grp ^
   --query 'Parameters[*].[ParameterName,ParameterValue]'
```

La sortie de cette opération ressemble à ceci (format JSON).

```
[
    [
        "audit_logs",
        "disabled"
    ],
    [
        "tls",
        "enabled"
```

```
    ],
    [
        "ttl_monitor",
        "enabled"
    ]
]
```

Modification des paramètres d'un groupe de paramètres de cluster Amazon DocumentDB

Vous pouvez modifier les paramètres d'un groupe de paramètres à l'aide de l'AWS Management Console ou de l'AWS CLI.

Using the AWS Management Console

Pour mettre à jour les paramètres d'un groupe de paramètres de cluster

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Groupes de paramètres.

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans le volet Parameter groups (Groupes de paramètres), choisissez le groupe de paramètres de cluster dont vous souhaitez mettre à jour les paramètres.
4. La page qui s’affiche montre les paramètres et leurs détails correspondants pour ce groupe de paramètres de cluster. Sélectionnez un paramètre à mettre à jour.
5. En haut à droite de la page, choisissez Edit (Modifier) pour modifier la valeur du paramètre. Pour de plus amples informations sur les types de paramètres de cluster, veuillez consulter Référence des paramètres du cluster Amazon DocumentDB.
6. Effectuez votre modification, puis choisissez Modify cluster parameter (Modifier le paramètre de cluster) pour enregistrer les modifications. Pour ignorer vos modifications, choisissez Annuler.

Using the AWS CLI

Pour modifier les paramètres d'un groupe de paramètres de cluster, utilisez l'opération `modify-db-cluster-parameter-group` avec les paramètres suivants :

- **`--db-cluster-parameter-group-name`** — Obligatoire. Nom du groupe de paramètres de cluster que vous voulez modifier.
- **`--parameters`** — Obligatoire. Le paramètre ou les paramètres que vous voulez modifier. Chaque saisie de paramètre doit inclure ce qui suit :
  - **`ParameterName`**— Le nom du paramètre que vous êtes en train de modifier.
  - **`ParameterValue`**— La nouvelle valeur de ce paramètre.
  - **`ApplyMethod`**— La manière dont vous souhaitez que les modifications soient appliquées à ce paramètre. Les valeurs autorisées sont `immediate` et `pending-reboot`.

> **Note**
>
> Les paramètres avec le `ApplyType` de `static` doivent avoir une `ApplyMethod` de `pending-reboot`.

Pour modifier les valeurs des paramètres d'un groupe de paramètres de cluster (AWS CLI)

L'exemple suivant modifie le paramètre `tls`.

1. Liste des paramètres et de leurs valeurs pour **`sample-parameter-group`**

   Pour Linux, macOS ou Unix :

   ```
   aws docdb describe-db-cluster-parameters \
       --db-cluster-parameter-group-name sample-parameter-group
   ```

   Pour Windows :

   ```
   aws docdb describe-db-cluster-parameters ^
       --db-cluster-parameter-group-name sample-parameter-group
   ```

   La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "Parameters": [
        {
            "Source": "system",
            "ApplyType": "static",
            "AllowedValues": "disabled,enabled",
            "ParameterValue": "enabled",
            "ApplyMethod": "pending-reboot",
            "DataType": "string",
            "ParameterName": "tls",
            "IsModifiable": true,
            "Description": "Config to enable/disable TLS"
        },
        {
            "Source": "user",
            "ApplyType": "dynamic",
            "AllowedValues": "disabled,enabled",
            "ParameterValue": "enabled",
            "ApplyMethod": "pending-reboot",
            "DataType": "string",
            "ParameterName": "ttl_monitor",
            "IsModifiable": true,
            "Description": "Enables TTL Monitoring"
        }
    ]
}
```

2. Modifier le paramètre **`tls`** afin que sa valeur soit **`disabled`**. Vous ne pouvez pas modifier la `ApplyMethod` car le `ApplyType` est `static`.

   Pour Linux, macOS ou Unix :

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group \
    --parameters
 "ParameterName"=tls,ParameterValue=disabled,ApplyMethod=pending-reboot"
```

   Pour Windows :

```
aws docdb modify-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name sample-parameter-group ^
```

```
    --parameters "ParameterName=tls,ParameterValue=disabled,ApplyMethod=pending-
reboot"
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "DBClusterParameterGroupName": "sample-parameter-group"
}
```

3. Patientez au moins 5 minutes.
4. Liste des valeurs du paramètre du **`sample-parameter-group`**.

   Pour Linux, macOS ou Unix :

```
aws docdb describe-db-cluster-parameters \
    --db-cluster-parameter-group-name sample-parameter-group
```

   Pour Windows :

```
aws docdb describe-db-cluster-parameters ^
    --db-cluster-parameter-group-name sample-parameter-group
```

   La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "Parameters": [
        {
            "ParameterName": "audit_logs",
            "ParameterValue": "disabled",
            "Description": "Enables auditing on cluster.",
            "Source": "system",
            "ApplyType": "dynamic",
            "DataType": "string",
            "AllowedValues": "enabled,disabled",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot"
        },
        {
            "ParameterName": "tls",
            "ParameterValue": "disabled",
            "Description": "Config to enable/disable TLS",
```

```
            "Source": "user",
            "ApplyType": "static",
            "DataType": "string",
            "AllowedValues": "disabled,enabled",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot"
        }
    ]
}
```

Comprendre les points de terminaison Amazon DocumentDB

Vous pouvez utiliser les points de terminaison Amazon DocumentDB (compatibles avec MongoDB) pour vous connecter à un cluster ou à une instance. Amazon DocumentDB possède trois types de points de terminaison différents, chacun ayant son propre objectif.

Rubriques

- Recherche des points de terminaison d'un cluster
- Recherche d'un point de terminaison de l'instance
- Connexion aux points de terminaison

Point de terminaison de cluster

Un point de terminaison de cluster est un point de terminaison d'un cluster Amazon DocumentDB qui se connecte à l'instance principale actuelle du cluster. Chaque cluster Amazon DocumentDB possède un point de terminaison de cluster unique et une instance principale. En cas de basculement, le point de terminaison du cluster est remappé vers la nouvelle instance principale.

Point de terminaison du lecteur

Un point de terminaison de lecteur est un point de terminaison d'un cluster Amazon DocumentDB qui se connecte à l'une des répliques disponibles pour ce cluster. Chaque cluster Amazon DocumentDB possède un point de terminaison lecteur. S'il existe plusieurs répliques, le point de terminaison du lecteur dirige chaque demande de connexion vers l'une des répliques Amazon DocumentDB.

Point de terminaison d'instance

Un point de terminaison d'une instance est un point de terminaison qui se connecte à cette instance spécifique. Chaque instance d'un cluster, qu'il s'agisse d'une instance principale ou

d'une instance de réplica, a son propre point de terminaison d'instance unique. Il est préférable de ne pas utiliser les points de terminaison d'instance dans votre application. En effet, ils peuvent modifier les rôles en cas de basculement, ce qui oblige à modifier le code dans votre application.

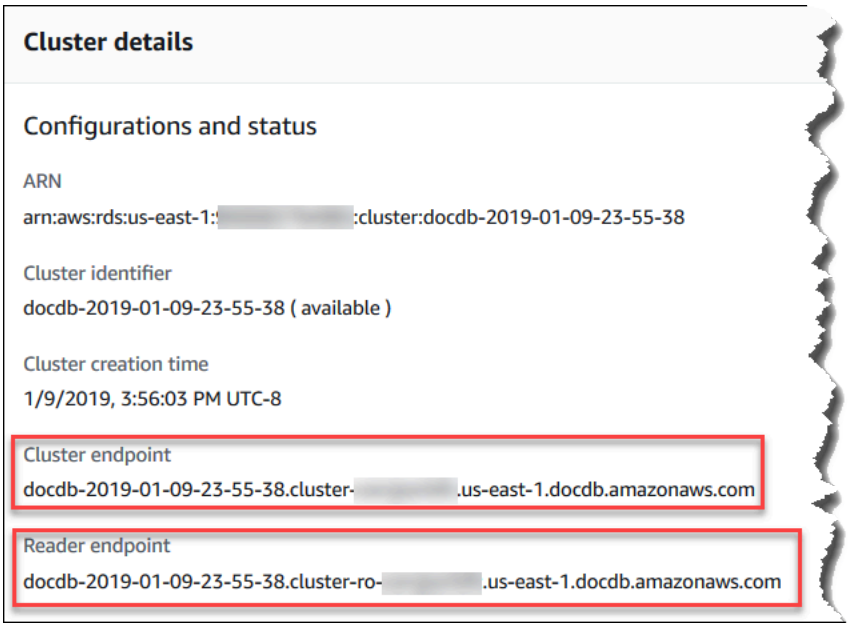
Recherche des points de terminaison d'un cluster

Vous pouvez trouver le point de terminaison du cluster et le point de terminaison du lecteur d'un cluster à l'aide de la console Amazon DocumentDB ouAWS CLI.

Using the AWS Management Console

Recherche des points de terminaison d'un cluster à l'aide de la console

1. Connectez-vous auAWS Management Console, et ouvrez la console Amazon DocumentDB à l'adressehttps://console.aws.amazon.com/docdb.
2. Dans le volet de navigation, choisissez clusters.
3. Dans la liste des clusters, choisissez le nom du cluster qui vous intéresse.
4. Faites défiler jusqu'à la rubrique Détails et recherchez le point de terminaison du cluster et le point de terminaison du lecteur.



5. Pour vous connecter à ce cluster, faites défiler jusqu'à la rubrique Connecter. Localisez la chaîne de connexion pour le shell mongo et une chaîne de connexion qui peuvent être utilisées dans le code d'application pour vous connecter à votre cluster.

Using the AWS CLI

Pour rechercher les points de terminaison du cluster et du lecteur pour votre cluster à l'aide de l'AWS CLI , exécutez la commande `describe-db-clusters` avec ces paramètres.

Paramètres

- **`--db-cluster-identifier`**—Facultatif. Spécifiez le cluster auquel renvoyer des points de terminaison. Si cet argument n'est pas spécifié, le nombre de points de terminaison renvoyés peut atteindre 100 de vos clusters.
- **`--query`**—Facultatif. Spécifiez les champs à afficher. Utile en réduisant la quantité de données que vous devez afficher pour trouver les points de terminaison. Si cela est omis, toutes les informations sur un cluster sont renvoyées.
- **`--region`**—Facultatif. Utilisez le paramètre `--region` pour spécifier la région à laquelle vous souhaitez appliquer la commande. S'il n'est pas spécifié, votre région par défaut est utilisée.

Example

L'exemple suivant renvoie le `DBClusterIdentifier`, le point de terminaison (point de terminaison du cluster) et le `ReaderEndpoint` pour `sample-cluster`.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-clusters \
   --region us-east-1 \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].[DBClusterIdentifier,Port,Endpoint,ReaderEndpoint]'
```

Pour Windows :

```
aws docdb describe-db-clusters ^
   --region us-east-1 ^
   --db-cluster-identifier sample-cluster ^
```

```
    --query 'DBClusters[*].[DBClusterIdentifier,Port,Endpoint,ReaderEndpoint]'
```

La sortie de cette opération ressemble à ceci (format JSON).

```
[
  [
     "sample-cluster",
     27017,
     "sample-cluster.cluster-corlsfccjozr.us-east-1.docdb.amazonaws.com",
     "sample-cluster.cluster-ro-corlsfccjozr.us-east-1.docdb.amazonaws.com"
  ]
]
```

Maintenant que vous disposez du point de terminaison du cluster, vous pouvez vous connecter au cluster à l'aide de `mongo` ou `mongodb`. Pour plus d'informations, veuillez consulter Connexion aux points de terminaison.

Recherche d'un point de terminaison de l'instance

Vous pouvez trouver le point de terminaison d'une instance à l'aide de la console Amazon DocumentDB ou duAWS CLI.

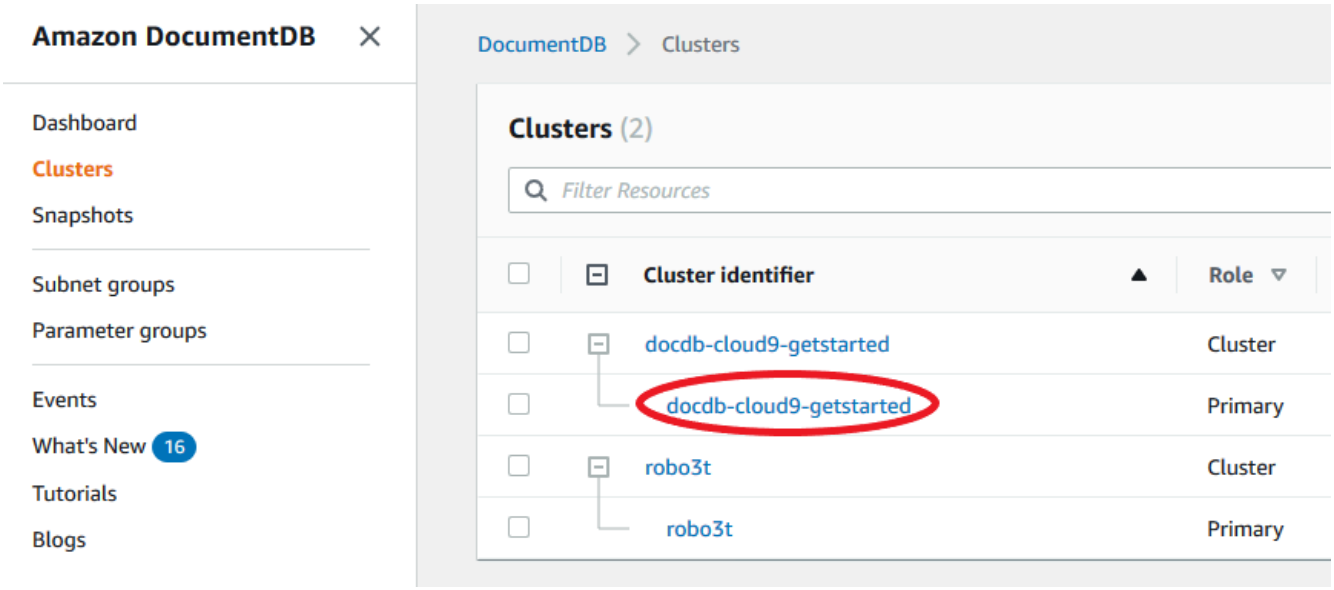
Using the AWS Management Console

Recherche d'un point de terminaison d'une instance à l'aide de la console

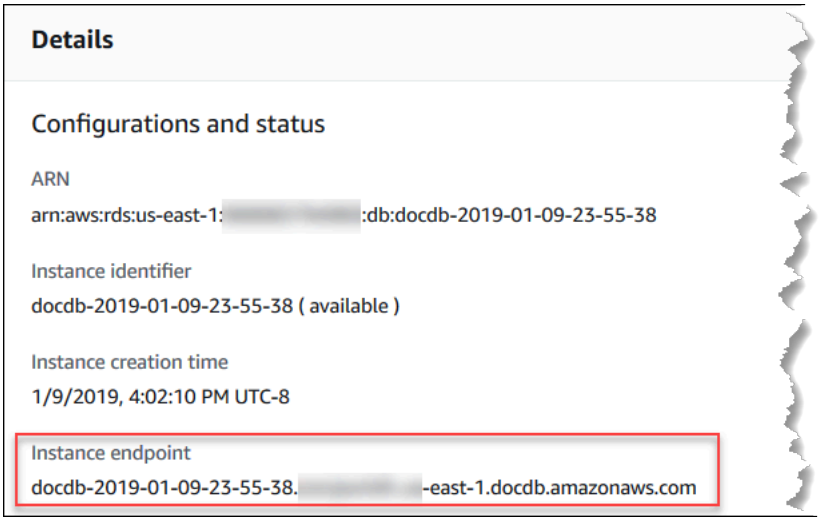
1. Connectez-vous auAWS Management Console, et ouvrez la console Amazon DocumentDB à l'adressehttps://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans la zone de navigation Clusters, vous verrez la colonneIdentifiant du cluster. Vos instances sont répertoriées sous des clusters, comme dans la capture d'écran ci-dessous.



4. Cochez la case située à gauche de l'instance qui vous intéresse.
5. Faites défiler jusqu'à la rubrique Détails, puis localisez le point de terminaison de l'instance.



6. Pour vous connecter à cette instance, faites défiler jusqu'à la rubrique Connecter. Localisez la chaîne de connexion pour le shell mongo et une chaîne de connexion qui peuvent être utilisées dans votre code d'application pour vous connecter à votre instance.

Using the AWS CLI

Pour rechercher le point de terminaison de l'instance à l'aide de la AWS CLI, exécutez la commande suivante avec ces arguments.

Arguments

- **`--db-instance-identifier`**—Facultatif. Spécifie l'instance à laquelle renvoyer le point de terminaison. Si cet argument n'est pas spécifié, le nombre de points de terminaison renvoyé peut atteindre 100 de vos instances.
- **`--query`**—Facultatif. Spécifiez les champs à afficher. Utile en réduisant la quantité de données que vous devez afficher pour trouver les points de terminaison. Si cela est omis, toutes les informations sur une instance sont renvoyées. Le champ `Endpoint` possède trois membres, son inclusion dans la requête comme dans l'exemple suivant renvoie donc les trois membres. Si vous êtes intéressé uniquement par certains des membres `Endpoint`, remplacez `Endpoint` dans la requête par les membres qui vous intéressent, comme dans le deuxième exemple.
- **`--region`**—Facultatif. Utilisez le paramètre `--region` pour spécifier la région à laquelle vous souhaitez appliquer la commande. S'il n'est pas spécifié, votre région par défaut est utilisée.

Example

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-instances \
    --region us-east-1 \
    --db-instance-identifier sample-cluster-instance \
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint]'
```

Pour Windows :

```
aws docdb describe-db-instances ^
    --region us-east-1 ^
    --db-instance-identifier sample-cluster-instance ^
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint]'
```

La sortie de cette opération ressemble à ceci (format JSON).

```
[
```

```
    [
        "sample-cluster-instance",
        {
            "Port": 27017,
            "Address": "sample-cluster-instance.corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
            "HostedZoneId": "Z2R2ITUGPM61AM"
        }
    ]
]
```

En réduisant les résultats pour éliminer la `HostedZoneId` du point de terminaison, vous pouvez modifier votre requête en spécifiant `Endpoint.Port` et `Endpoint.Address`.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-instances \
    --region us-east-1 \
    --db-instance-identifier sample-cluster-instance \
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint.Port,Endpoint.Address]'
```

Pour Windows :

```
aws docdb describe-db-instances ^
    --region us-east-1 ^
    --db-instance-identifier sample-cluster-instance ^
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint.Port,Endpoint.Address]'
```

La sortie de cette opération ressemble à ceci (format JSON).

```
[
    [
        "sample-cluster-instance",
        27017,
        "sample-cluster-instance.corcjozrlsfc.us-east-1.docdb.amazonaws.com"
    ]
]
```

Maintenant que vous disposez du point de terminaison de l'instance, vous pouvez vous connecter à l'instance à l'aide de `mongo` ou `mongodb`. Pour plus d'informations, veuillez consulter Connexion aux points de terminaison.

Connexion aux points de terminaison

Lorsque vous avez votre point de terminaison, cluster ou instance, vous pouvez vous y connecter à l'aide du shell mongo ou d'une chaîne de connexion.

Connexion à l'aide du shell mongo

Utilisez la structure suivante pour construire la chaîne dont vous avez besoin pour vous connecter à votre cluster ou instance à l'aide du shell mongo :

```
mongo \
    --ssl \
    --host Endpoint:Port \
    --sslCAFile global-bundle.pem \
    --username UserName \
    --password Password
```

Exemples de shell **mongo**

Connexion à un cluster :

```
mongo \
    --ssl \
    --host sample-cluster.corcjozrlsfc.us-east-1.docdb.amazonaws.com:27017 \
    --sslCAFile global-bundle.pem \
    --username UserName \
    --password Password
```

Connexion à une instance :

```
mongo \
    --ssl \
    --host sample-cluster-instance.corcjozrlsfc.us-east-1.docdb.amazonaws.com:27017 \
    --sslCAFile global-bundle.pem \
    --username UserName \
    --password Password
```

Connexion à l'aide d'une chaîne de connexion

Utilisez la structure suivante pour construire la chaîne de connexion dont vous avez besoin pour vous connecter à votre cluster ou instance.

```
mongodb://UserName:Password@endpoint:port?replicaSet=rs0&ssl_ca_certs=global-
bundle.pem
```

Exemples de chaînes de connexion

Connexion à un cluster :

```
mongodb://UserName:Password@sample-cluster.cluster-corlsfccjozr.us-
east-1.docdb.amazonaws.com:27017?replicaSet=rs0&ssl_ca_certs=global-bundle.pem
```

Connexion à une instance :

```
mongodb://UserName:Password@sample-cluster-instance.cluster-corlsfccjozr.us-
east-1.docdb.amazonaws.com:27017?replicaSet=rs0&ssl_ca_certs=global-bundle.pem
```

Comprendre les noms de ressources Amazon (ARN) Amazon DocumentDB

Les ressources que vous créez dans AWS sont toutes identifiées de manière unique par un Amazon Resource Name (ARN). Pour certaines opérations Amazon DocumentDB (compatibles avec MongoDB), vous devez identifier de manière unique une ressource Amazon DocumentDB en spécifiant son ARN. Par exemple, lorsque vous ajoutez une balise à une ressource, vous devez fournir l'ARN de la ressource.

Rubriques

- Création d'un ARN pour une ressource Amazon DocumentDB
- Trouver un ARN de ressource Amazon DocumentDB

Création d'un ARN pour une ressource Amazon DocumentDB

Vous pouvez créer un ARN pour une ressource Amazon DocumentDB en utilisant la syntaxe suivante. Amazon DocumentDB partage le format des ARNS d'Amazon Relational Database Service (Amazon RDS). Les ARN Amazon DocumentDB contiennent `rds` et non. `docdb`

`arn:aws:rds:`*`region`*`:`*`account_number`*`:`*`resource_type`*`:`*`resource_id`*

Nom de la région	Région	Zones de disponibilité (calcul)
USA Est (Ohio)	`us-east-2`	3
USA Est (Virginie du Nord)	`us-east-1`	6
USA Ouest (Oregon)	`us-west-2`	4
Amérique du Sud (São Paulo)	`sa-east-1`	3
Asie-Pacifique (Hong Kong)	`ap-east-1`	3
Asie-Pacifique (Hyderabad)	`ap-south-2`	3
Asie-Pacifique (Mumbai)	`ap-south-1`	3
Asie-Pacifique (Séoul)	`ap-northeast-2`	4
Asie-Pacifique (Singapour)	`ap-southeast-1`	3
Asie-Pacifique (Sydney)	`ap-southeast-2`	3
Asie-Pacifique (Tokyo)	`ap-northeast-1`	3
Canada (Centre)	`ca-central-1`	3
Région Chine (Beijing)	`cn-north-1`	3
Chine (Ningxia)	`cn-northwest-1`	3
Europe (Francfort)	`eu-central-1`	3

Nom de la région	Région	Zones de disponibilité (calcul)
Europe (Irlande)	`eu-west-1`	3
Europe (Londres)	`eu-west-2`	3
Europe (Milan)	`eu-south-1`	3
Europe (Paris)	`eu-west-3`	3
Moyen-Orient (EAU)	`me-central-1`	3
AWS GovCloud (US-Ouest)	`us-gov-west-1`	3
AWS GovCloud (USA Est)	`us-gov-east-1`	3

> **Note**
>
> L'architecture Amazon DocumentDB sépare le stockage et le calcul. Pour la couche de stockage, Amazon DocumentDB réplique six copies de vos données dans trois zones de AWS disponibilité (AZ). Les zones de disponibilité répertoriées dans le tableau ci-dessus représentent le nombre de zones que vous pouvez utiliser dans une région donnée afin d'approvisionner des instances de calcul. Par exemple, si vous lancez un cluster Amazon DocumentDB dans ap-northeast-1, votre stockage sera répliqué de six manières sur trois zones de disponibilité, mais vos instances de calcul ne seront disponibles que dans deux zones de disponibilité.

Le tableau suivant indique le format que vous devez utiliser lors de la création d'un ARN pour une ressource Amazon DocumentDB spécifique. Amazon DocumentDB partage le format d'Amazon RDS ARNS. Les ARN Amazon DocumentDB contiennent `rds` et non. `docdb`

Type de ressource	Format ARN/Exemple
Instance (db)	`arn:aws:rds:` *`region`*`:`*`account_number`*`:db:`*`resource_id`*

Type de ressource	Format ARN/Exemple
	`arn:aws:rds:us-east-1:` *1234567890* `:db:`*sample-db-instance*
Cluster (`cluster`)	`arn:aws:rds:` *region*`:`*account_number* `:cluster:` *resource_id* `arn:aws:rds:us-east-1:` *1234567890* `:cluster:` *sample-db-cluster*
Groupe de paramètres du cluster (`cluster-pg`)	`arn:aws:rds:` *region*`:`*account_number* `:cluster-pg:` *resource_id* `arn:aws:rds:us-east-1:` *1234567890* `:cluster-pg:` *sample-db-cluster-parameter-group*
Groupe de sécurité (`secgrp`)	`arn:aws:rds:` *region*`:`*account_number* `:secgrp:`*resource_id* `arn:aws:rds:us-east-1:` *1234567890* `:secgrp:`*sample-public-secgrp*
Instantané du cluster (`cluster-snapshot`)	`arn:aws:rds:` *region*`:`*account_number* `:cluster-snapshot:` *resource_id* `arn:aws:rds:us-east-1:` *1234567890* `:cluster-snapshot:` *sample-db-cluster-snapshot*
Groupe de sous-réseaux (`subgrp`)	`arn:aws:rds:` *region*`:`*account_number* `:subgrp:`*resource_id* `arn:aws:rds:us-east-1:` *1234567890* `:subgrp:`*sample-subnet-10*

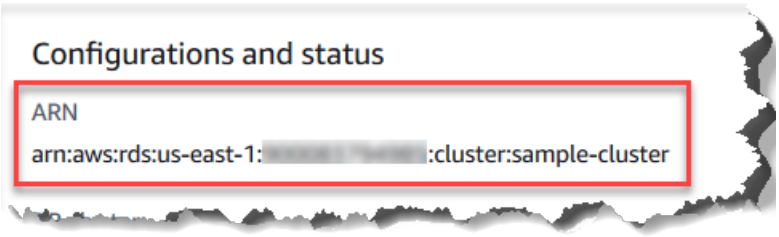
Trouver un ARN de ressource Amazon DocumentDB

Vous pouvez trouver l'ARN d'une ressource Amazon DocumentDB en utilisant le AWS Management Console ou le. AWS CLI

Using the AWS Management Console

Pour trouver un ARN à l'aide de la console, accédez à la ressource pour laquelle vous souhaitez obtenir un ARN, puis consultez les détails de cette ressource.

Par exemple, vous pouvez obtenir l'ARN pour un cluster dans le volet Détails pour le cluster, comme illustré dans la capture d'écran suivante.



Using the AWS CLI

Pour obtenir un ARN en utilisant le AWS CLI pour une ressource Amazon DocumentDB particulière, utilisez l'`describe`opération pour cette ressource. Le tableau suivant indique chaque AWS CLI opération et la propriété ARN utilisée avec l'opération pour obtenir un ARN.

AWS CLI Commande	Propriété d'ARN
`describe-db-instances`	`DBInstanceArn`
`describe-db-clusters`	`DBClusterArn`
`describe-db-parameter-groups`	`DBParameterGroupArn`
`describe-db-cluster-parameter-groups`	`DBClusterParameterGroupArn`
`describe-db-security-groups`	`DBSecurityGroupArn`
`describe-db-snapshots`	`DBSnapshotArn`
`describe-db-cluster-snapshots`	`DBClusterSnapshotArn`

AWS CLI Commande	Propriété d'ARN
`describe-db-subnet-groups`	`DBSubnetGroupArn`

Example - Recherche de l'ARN d'un cluster

L' AWS CLI opération suivante permet de trouver l'ARN du cluster`sample-cluster`.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].DBClusterArn'
```

Pour Windows :

```
aws docdb describe-db-clusters ^
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].DBClusterArn'
```

La sortie de cette opération ressemble à ceci (format JSON).

```
[
    "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster"
]
```

Example - Recherche des ARN pour plusieurs groupes de paramètres

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-cluster-parameter-groups \
   --query 'DBClusterParameterGroups[*].DBClusterParameterGroupArn'
```

Pour Windows :

```
aws docdb describe-db-cluster-parameter-groups ^
   --query 'DBClusterParameterGroups[*].DBClusterParameterGroupArn'
```

La sortie de cette opération ressemble à ceci (format JSON).

```
[
    "arn:aws:rds:us-east-1:123456789012:cluster-pg:custom3-6-param-grp",
    "arn:aws:rds:us-east-1:123456789012:cluster-pg:default.aurora5.6",
    "arn:aws:rds:us-east-1:123456789012:cluster-pg:default.docdb3.6"
]
```

Balisage des ressources Amazon DocumentDB

Vous pouvez utiliser les balises Amazon DocumentDB (compatibles avec MongoDB) pour ajouter des métadonnées à vos ressources Amazon DocumentDB. Ces balises peuvent être utilisées avecAWS Identity and Access Managementpolitiques (IAM) pour gérer l'accès aux ressources Amazon DocumentDB et pour contrôler les actions qui peuvent être appliquées aux ressources. Vous pouvez également utiliser ces balises pour suivre les coûts en regroupant les dépenses pour des ressources balisées de la même façon.

Vous pouvez baliser les ressources Amazon DocumentDB suivantes :

- Clusters
- instances
- Instantanés
- Instantanés de cluster
- Groupes de paramètres
- Groupes de paramètres de cluster
- Groupes de sécurité
- Groupes de sous-réseaux

Présentation des balises de ressources Amazon DocumentDB

Une balise Amazon DocumentDB est une paire nom-valeur que vous définissez et associez à une ressource Amazon DocumentDB. Le nom s'appelle la clé. Fournir une valeur pour la clé est facultatif. Vous pouvez utiliser des balises pour attribuer des informations arbitraires à une ressource Amazon DocumentDB. Vous pouvez utiliser une clé de balise, par exemple, pour définir une catégorie, et la valeur de balise peut être un élément de cette catégorie. Par exemple, vous pouvez

définir une clé de balise de`project`et une valeur de balise de`Salix`, indiquant que la ressource Amazon DocumentDB est affectée au projet Salix. Vous pouvez également utiliser des balises pour désigner les ressources Amazon DocumentDB comme étant utilisées à des fins de test ou de production à l'aide d'une clé telle que`environment=test`ou`environment=production`. Nous vous recommandons d'utiliser un ensemble cohérent de clés de balise pour faciliter le suivi des métadonnées associées aux ressources Amazon DocumentDB.

Vous pouvez utiliser des balises pour organiser votre facture AWS afin de refléter votre propre structure de coût. Pour ce faire, inscrivez-vous pour obtenir votre facture Compte AWS avec les valeurs de clé de balise incluses. Ensuite, pour voir le coût de vos ressources combinées, organisez vos informations de facturation en fonction des ressources possédant les mêmes valeurs de clé de balise. Par exemple, vous pouvez baliser plusieurs ressources avec un nom d'application spécifique, puis organiser vos informations de facturation pour afficher le coût total de cette application dans plusieurs services. Pour plus d'informations, voirUtilisation des balises de répartition des coûtsdans leAWSGuide de l'utilisateur sur la facturation et la gestion des coûts.

Chaque ressource Amazon DocumentDB possède un ensemble de balises, qui contient toutes les balises attribuées à cette ressource. Un ensemble de balises peut contenir jusqu'à dix balises ou n'en contenir aucune. Si vous ajoutez une balise à une ressource Amazon DocumentDB qui possède la même clé qu'une balise existante sur la ressource, la nouvelle valeur remplace l'ancienne valeur.

AWS n'applique aucune signification sémantique aux balises ; celles-ci sont interprétées strictement comme des chaînes de caractères. Amazon DocumentDB peut définir des balises sur une instance ou d'autres ressources Amazon DocumentDB, en fonction des paramètres que vous utilisez lors de la création de la ressource. Par exemple, Amazon DocumentDB peut ajouter une balise indiquant qu'une instance est destinée à la production ou aux tests.

Vous pouvez ajouter une balise à un instantané, toutefois, votre facture ne reflètera pas ce groupement.

Vous pouvez utiliserAWS Management Consoleou leAWS CLIpour ajouter, répertorier et supprimer des balises sur les ressources Amazon DocumentDB. Lorsque vous utilisez la AWS CLI, vous devez fournir l'Amazon Resource Name (ARN) pour la ressource avec laquelle vous souhaitez travailler. Pour plus d'informations sur les ARN Amazon DocumentDB, consultezComprendre les noms de ressources Amazon (ARN) Amazon DocumentDB.

Restrictions liées aux balises

Les contraintes suivantes s'appliquent aux balises Amazon DocumentDB :

- Nombre maximum de balises par ressource : 10
- Longueur de Key maximale - 128 caractères Unicode.
- Longueur de valeur maximale - 256 caractères Unicode.
- Caractères valables pour Clé et Valeur - lettres majuscules et minuscules UTF-8, chiffres, espace et les caractères suivants : `_ . : / = + -` et @ (Java regex : `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"`)
- Les clés et valeurs de balise sont sensibles à la casse.
- Le préfixe `aws:` ne peut pas être utilisé pour les valeurs ou clés de balise ; il est réservé à AWS.

Ajouter et mettre à jour des balises sur une ressource Amazon DocumentDB

Vous pouvez ajouter jusqu'à 10 balises à une ressource à l'aide duAWS Management Consoleou leAWS CLI.

Using the AWS Management Console

Le processus d'ajout d'une balise à une ressource est semblable, quelle que soit la ressource à laquelle vous ajoutez la balise. Dans cet exemple, vous ajoutez une balise à un cluster.

Ajouter ou mettre à jour des balises à un cluster à l'aide de la console

1. Connectez-vous auAWS Management Console, et ouvrez la console Amazon DocumentDB à l'adressehttps://console.aws.amazon.com/docdb.
2. Dans le volet de navigation, choisissez clusters.
3. Choisissez le nom du cluster auquel vous souhaitez ajouter des balises.
4. Faites défiler jusqu'à la section Balises, puis choisissez Modifier.
5. Pour chaque balise que vous souhaitez ajouter à cette ressource, procédez comme suit :
    a. Pour ajouter une nouvelle balise, saisissez le nom de la balise dans la case Clé. Pour modifier la valeur d'une balise, recherchez le nom de la balise dans la colonne Clé.
    b. Pour doter la balise d'une valeur nouvelle ou mise à jour, saisissez une valeur pour la balise dans la case Valeur.
    c. Pour ajouter davantage de balises, choisissez Ajouter. Lorsque vous avez terminé, choisissez Sauvegarder.

Using the AWS CLI

Le processus d'ajout d'une balise à une ressource est semblable, quelle que soit la ressource à laquelle vous ajoutez la balise. Dans cet exemple, vous ajoutez trois balises à un cluster. La deuxième balise, `key2`, n'a pas de valeur.

Avec l'opération AWS CLI, utilisez ces paramètres `add-tags-to-resource`.

Paramètres

- **`--resource-name`**: l'ARN de la ressource Amazon DocumentDB à laquelle vous souhaitez ajouter des balises.
- **`--tags`**—Liste les balises (paire clé-valeur) que vous souhaitez ajouter à cette ressource au format`Key=`*`key-name`*`,Value=`*`tag-value`*.

Example

Pour Linux, macOS ou Unix :

```
aws docdb add-tags-to-resource \
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \
    --tags Key=key1,Value=value1 Key=key2 Key=key3,Value=value3
```

Pour Windows :

```
aws docdb add-tags-to-resource ^
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \
    --tags Key=key1,Value=value1 Key=key2 Key=key3,Value=value3
```

L'opération `add-tags-to-resource` ne produit aucun résultat. Pour consulter les résultats de l'opération, utilisez l'opération `list-tags-for-resource`.

Répertorier des balises sur une ressource Amazon DocumentDB

Vous pouvez utiliserAWS Management Consoleou leAWS CLIpour obtenir la liste des balises d'une ressource Amazon DocumentDB.

Using the AWS Management Console

Le processus pour établir une liste de balises pour une ressource est semblable, quelle que soit la ressource à laquelle vous ajoutez la balise. Dans cet exemple, vous établissez la liste des balises pour un cluster.

Établir la liste des balises pour un cluster à l'aide de la console

1. Ouvrez la console Amazon DocumentDB à l'adressehttps://console.aws.amazon.com/docdb.
2. Dans le volet de navigation, choisissez clusters.
3. Choisissez le nom du cluster pour lequel vous souhaitez établir la liste des balises.
4. Pour afficher la liste des balises pour cette ressource, faites défiler vers le bas jusqu'à la section Balises.

Using the AWS CLI

Le processus pour établir une liste de balises pour une ressource est semblable, quelle que soit la ressource pour laquelle vous établissez la liste des balises. Dans cet exemple, vous établissez la liste des balises pour un cluster.

Avec l'opération AWS CLI, utilisez ces paramètres `list-tags-for-resource`.

Paramètres

- **`--resource-name`** : obligatoire. L'ARN de la ressource Amazon DocumentDB pour laquelle vous souhaitez répertorier les balises.

Example

Pour Linux, macOS ou Unix :

```
aws docdb list-tags-for-resource \
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster
```

Pour Windows :

```
aws docdb list-tags-for-resource ^
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "TagList": [
        {
            "Key": "key1",
            "Value": "value1"
        },
        {
            "Key": "key2",
            "Value": ""
        },
        {
            "Key": "key3",
            "Value": "value3"
        }
    ]
}
```

Supprimer des balises d'une ressource Amazon DocumentDB

Vous pouvez utiliserAWS Management Consoleou leAWS CLIpour supprimer les balises des ressources Amazon DocumentDB.

Using the AWS Management Console

Le processus de suppression de balises dans une ressource est semblable, quelle que soit la ressource dans laquelle vous voulez supprimer la balise. Dans cet exemple, vous supprimez des balises dans un cluster.

Pour supprimer les balises d'un cluster à l'aide de la console

1. Ouvrez la console Amazon DocumentDB à l'adressehttps://console.aws.amazon.com/docdb.
2. Dans le volet de navigation, choisissez clusters.
3. Choisissez le nom du cluster pour lequel vous souhaitez supprimer des balises.
4. Faites défiler jusqu'à la section Balises, puis choisissez Modifier.
5. Si vous souhaitez supprimer toutes les balises de cette ressource, choisissez Tout supprimer. Dans le cas contraire, pour chaque balise que vous souhaitez supprimer dans cette ressource, procédez comme suit :

a. Recherchez le nom de la balise dans la colonne Clé.

b. Choisissez Supprimer sur la même ligne que la clé de la balise.

c. Lorsque vous avez terminé, choisissez Save (Sauvegarder).

Using the AWS CLI

Le processus de suppression d'une balise dans une ressource est semblable, quelle que soit la ressource dans laquelle vous voulez supprimer la balise. Dans cet exemple, vous supprimez une balise dans un cluster.

Avec l'opération AWS CLI, utilisez ces paramètres `remove-tags-from-resource`.

- **`--resource-name`** : obligatoire. L'ARN de la ressource Amazon DocumentDB dont vous souhaitez supprimer les balises.
- **`--tag-keys`** : obligatoire. Liste des clés des balises que vous voulez supprimer dans cette ressource.

Example

Pour Linux, macOS ou Unix :

```
aws docdb remove-tags-from-resource \
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \
    --tag-keys key1 key3
```

Pour Windows :

```
aws docdb remove-tags-from-resource ^
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \
    --tag-keys key1 key3
```

L'opération `removed-tags-from-resource` ne produit aucun résultat. Pour consulter les résultats de l'opération, utilisez l'opération `list-tags-for-resource`.

Gestion d'Amazon DocumentDB

Amazon DocumentDB effectue régulièrement la maintenance des ressources Amazon DocumentDB. La maintenance implique le plus souvent des mises à jour du moteur de base de données (maintenance de cluster) ou du système d'exploitation sous-jacent de l'instance (maintenance de l'instance). Les mises à jour du moteur de base de données sont des correctifs obligatoires et incluent des correctifs de sécurité, des corrections de bogues et des améliorations du moteur de base de données. Les mises à jour du système d'exploitation incluent souvent des correctifs de sécurité. Bien que les correctifs du système d'exploitation soient facultatifs, nous vous recommandons de les appliquer à vos instances Amazon DocumentDB dès qu'ils sont disponibles.

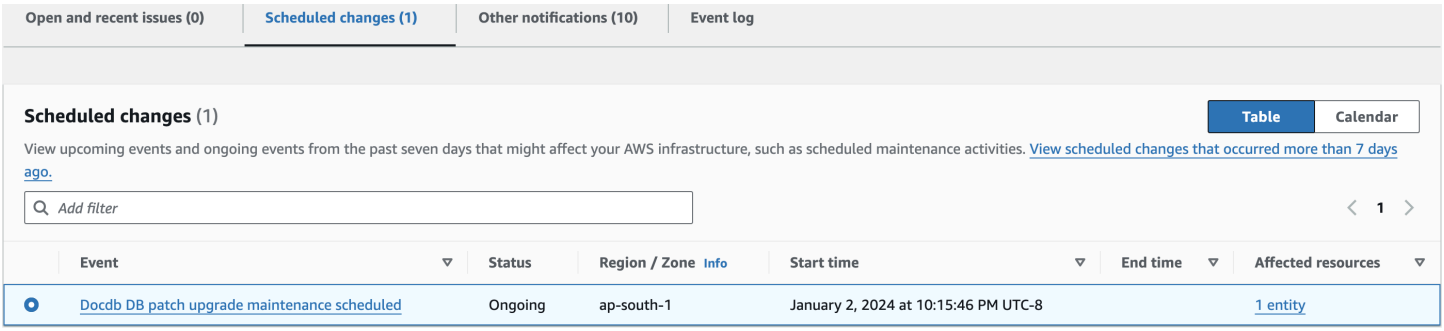
Les correctifs du moteur de base de données nécessitent que vous mettiez vos clusters Amazon DocumentDB hors ligne pendant une courte période. Une fois disponibles, ces correctifs sont automatiquement programmés pour s'appliquer lors d'une prochaine période de maintenance planifiée de votre cluster Amazon DocumentDB.

La maintenance du cluster et celle des instances ont leurs propres fenêtres de maintenance. Les modifications de cluster et d'instance que vous avez choisi de ne pas appliquer immédiatement sont également appliquées pendant la fenêtre de maintenance. Par défaut, lorsque vous créez un cluster, Amazon DocumentDB attribue une fenêtre de maintenance à la fois au cluster et à chaque instance individuelle. Vous pouvez choisir la fenêtre de maintenance lors de la création d'un cluster ou d'une instance. Vous pouvez également modifier les fenêtres de maintenance à tout moment en fonction de vos planifications ou pratiques métier. Il est généralement conseillé de choisir des fenêtres de maintenance qui limitent l'impact de la maintenance sur votre application (par exemple, le soir ou le week-end). Ces conseils sont extrêmement contextuels et dépendent du type d'application et des modèles d'utilisation que vous rencontrez.

Rubriques

Notifications relatives aux correctifs du moteur Amazon DocumentDB

Vous recevrez des notifications de maintenance pour les correctifs de moteur de base de données requis par le biais d'événements de santé dans le AWS Health Dashboard (AHD) de la AWS console et par e-mail. Lorsqu'un correctif de maintenance du moteur Amazon DocumentDB est disponible dans une AWS région donnée, tous les comptes utilisateurs Amazon DocumentDB concernés de la région reçoivent un AHD et une notification par e-mail pour chaque version d'Amazon DocumentDB affectée par le correctif. Vous pouvez consulter ces notifications dans la section Modifications planifiées de l'AHD dans la AWS console. La notification contiendra des détails sur le calendrier de disponibilité des correctifs, le calendrier d'application automatique, la liste des clusters concernés et les notes de mise à jour. Cette notification sera également envoyée par e-mail à l'adresse e-mail de l'utilisateur root du AWS compte.



Une fois que vous aurez reçu cette notification, vous pourrez choisir d'appliquer automatiquement ces correctifs de moteur à vos clusters Amazon DocumentDB avant la date d'application automatique prévue. Vous pouvez également attendre que les correctifs du moteur soient appliqués automatiquement lors d'une prochaine fenêtre de maintenance (option par défaut).

> **Note**
>
> Le statut de la notification dans l'AHD sera défini sur « En cours » jusqu'à ce qu'un nouveau correctif du moteur Amazon DocumentDB contenant une nouvelle version du correctif soit publié.
>
> Une fois le correctif du moteur appliqué à votre cluster Amazon DocumentDB, la version du correctif du moteur du cluster sera mise à jour pour refléter la version indiquée dans la notification. Vous pouvez exécuter la `db.runCommand({getEngineVersion: 1})` commande pour vérifier cette mise à jour.

AWS Health s'intègre également à Amazon EventBridge , qui utilise des événements pour créer des applications évolutives axées sur les événements et s'intègre à plus de 20 cibles AWS Lambda, dont Amazon Simple Queue Service (SQS), entre autres. Vous pouvez utiliser le code `AWS_DOCDB_DB_PATCH_UPGRADE_MAINTENANCE_SCHEDULED` d'événement pour configurer Amazon EventBridge avant que les correctifs du moteur ne soient disponibles. Vous pouvez le configurer EventBridge pour répondre à l'événement et exécuter automatiquement des actions telles que la capture des informations relatives à l'événement, le lancement d'événements supplémentaires, l'envoi de notifications via des canaux supplémentaires tels que les notifications push au AWS Console Mobile Application, et la prise de mesures correctives ou autres lorsqu'un correctif du moteur Amazon DocumentDB est disponible dans votre région.

Dans les rares cas où Amazon DocumentDB annule un correctif moteur, vous recevrez une notification AHD ainsi qu'un e-mail vous informant de l'annulation. Par conséquent, vous pouvez utiliser le code `AWS_DOCDB_DB_PATCH_UPGRADE_MAINTENANCE_CANCELLED` d'événement pour configurer Amazon EventBridge afin qu'il réponde à cet événement. Consultez le guide de EventBridge l'utilisateur Amazon pour en savoir plus sur l'utilisation EventBridge des règles Amazon.

Affichage des actions de maintenance Amazon DocumentDB en attente

Vous pouvez voir si une mise à jour de maintenance est disponible pour votre cluster en utilisant le AWS Management Console ou le AWS CLI.

Si une mise à jour est disponible, vous pouvez procéder de l'une des manières suivantes :

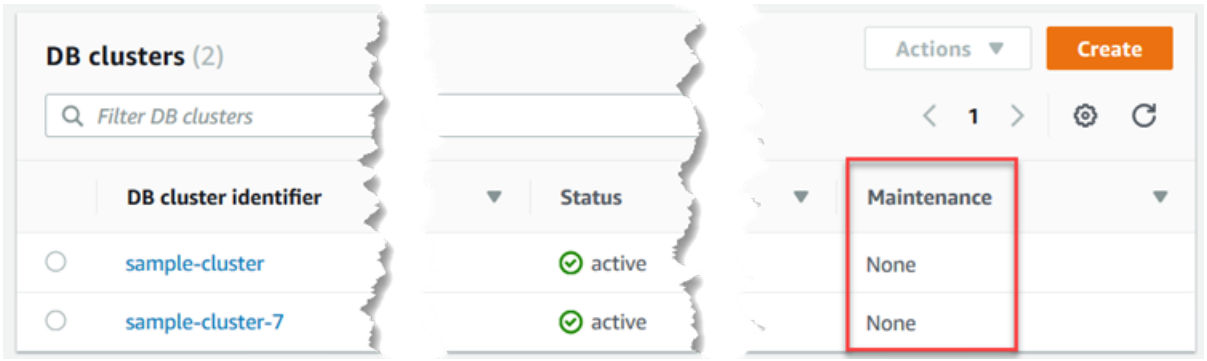
- Reportez une action de maintenance actuellement planifiée pour la prochaine fenêtre de maintenance (pour les correctifs du système d'exploitation uniquement).
- Appliquer immédiatement les actions de maintenance.
- Planifier le début des actions de maintenance au cours de votre prochaine fenêtre de maintenance.

> **Note**
>
> Si vous ne prenez aucune mesure, les actions de maintenance requises, telles que les correctifs du moteur, seront appliquées automatiquement lors d'une prochaine fenêtre de maintenance planifiée.

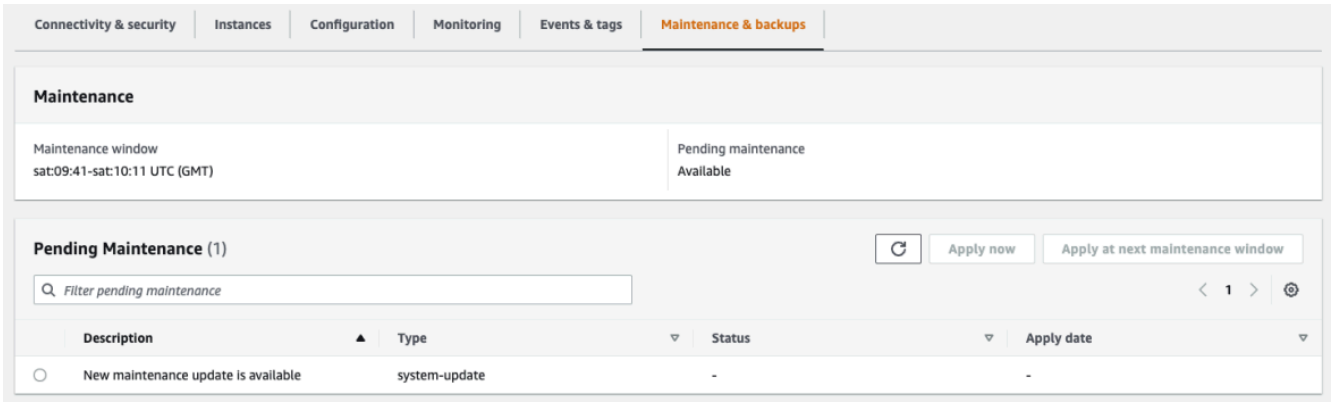
Le réglage de la fenêtre de maintenance détermine quand les opérations en attente démarrent, mais il ne limite pas la durée d'exécution totale de ces opérations.

Using the AWS Management Console

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.
3. Si une mise à jour est disponible, elle est indiquée par le mot Available, Required ou Next Window dans la colonne Maintenance du cluster sur la console Amazon DocumentDB, comme indiqué ici :



4. Pour effectuer une action, choisissez le cluster pour en afficher les détails, puis choisissez Maintenance et sauvegardes. Les éléments de maintenance en attente apparaissent.



Using the AWS CLI

Utilisez l' AWS CLI opération suivante pour déterminer les actions de maintenance en attente. Le résultats obtenu ici n'indique aucune action de maintenance en attente.

```
aws docdb describe-pending-maintenance-actions
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "PendingMaintenanceActions": []
}
```

Appliquer les mises à jour du moteur Amazon DocumentDB

Avec Amazon DocumentDB, vous pouvez choisir à quel moment appliquer les opérations de maintenance. Vous pouvez décider quand Amazon DocumentDB applique les mises à jour à l'aide du AWS Management Console ou. AWS CLI

Utilisez les procédures décrites dans cette rubrique pour mettre immédiatement à niveau ou planifier une mise à niveau pour votre cluster.

Using the AWS Management Console

Vous pouvez utiliser la console pour gérer les mises à jour de vos clusters Amazon DocumentDB.

Pour gérer une mise à jour pour un cluster

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.
3. Dans la liste des clusters, choisissez le bouton en regard du nom du cluster auquel vous voulez appliquer l'opération de maintenance.
4. Dans le menu Actions, choisissez l'une des options suivantes :

   - Mettre à niveau maintenant pour exécuter immédiatement les tâches de maintenance en attente.
   - Mettre à niveau lors du créneau suivant pour exécuter les tâches de maintenance en attente pendant le créneau de maintenance suivant du cluster.

   Vous pouvez également cliquer sur Appliquer maintenant ou sur Appliquer dans la fenêtre de maintenance suivante dans la section Maintenance en attente de l'onglet Maintenance et sauvegardes du cluster (voir Utilisation de la AWS Management Console section précédente).

> **Note**
>
> S'il n'y a aucune tâche de maintenance en attente, toutes les options précédentes sont inactives.

Using the AWS CLI

Pour appliquer une mise à jour en attente à un cluster, utilisez l'`apply-pending-maintenance-action` AWS CLI opération.

Paramètres

- **`--resource-identifier`**—Le nom de ressource Amazon (ARN) Amazon DocumentDB de la ressource à laquelle s'applique l'action de maintenance en attente.
- **`--apply-action`**: action de maintenance en attente à appliquer à cette ressource.

  Valeurs valables : `system-update` et `db-upgrade`.
- **`--opt-in-type`**—Une valeur qui spécifie le type de demande d'opt-in ou qui annule une demande d'opt-in. Un type demande de confirmation de l'acceptation de type `immediate` ne peut pas être annulée.

  Valeurs valides :
  - `immediate`—Appliquez immédiatement l'action de maintenance.
  - `next-maintenance`—Appliquez l'action de maintenance lors de la fenêtre de maintenance suivante pour la ressource.
  - `undo-opt-in`—Annulez toutes les demandes d'`next-maintenance`opt-in existantes.

Example

Pour Linux, macOS ou Unix :

```
aws docdb apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-east-1:123456789012:db:docdb \
    --apply-action system-update \
    --opt-in-type immediate
```

Pour Windows :

```
aws docdb apply-pending-maintenance-action ^
    --resource-identifier arn:aws:rds:us-east-1:123456789012:db:docdb ^
    --apply-action system-update ^
    --opt-in-type immediate
```

Pour renvoyer une liste de ressources dont au moins une mise à jour est en attente, utilisez l'`describe-pending-maintenance-actions` AWS CLI opération.

Example

Pour Linux, macOS ou Unix :

```
aws docdb describe-pending-maintenance-actions \
    --resource-identifier arn:aws:rds:us-east-1:001234567890:db:docdb
```

Pour Windows :

```
aws docdb describe-pending-maintenance-actions ^
    --resource-identifier arn:aws:rds:us-east-1:001234567890:db:docdb
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "PendingMaintenanceActions": [
        {
            "ResourceIdentifier": "arn:aws:rds:us-
east-1:001234567890:cluster:sample-cluster",
            "PendingMaintenanceActionDetails": [
                {
                    "Action": "system-update",
                    "CurrentApplyDate": "2019-01-11T03:01:00Z",
                    "Description": "db-version-upgrade",
                    "ForcedApplyDate": "2019-01-18T03:01:00Z",
                    "AutoAppliedAfterDate": "2019-01-11T03:01:00Z"
                }
            ]
        }
    ]
}
```

Vous pouvez également renvoyer une liste de ressources pour un cluster en spécifiant le `--filters` paramètre de l'`describe-pending-maintenance-actions` AWS CLI opération. Le format de l'opération `--filters` est `Name=`*`filter-name`*`,Values=`*`resource-id`*`,...`.

`db-cluster-id`est la valeur acceptable pour le `Name` paramètre du filtre. Cette valeur accepte une liste d'identifiants de cluster ou d'ARN. La liste renvoyée inclut uniquement les actions de maintenance en attente pour les clusters identifiés par ces identifiants ou ARN.

L'exemple suivant renvoie les actions de maintenance en attente pour les clusters `sample-cluster1` et `sample-cluster2`.

Example

Pour Linux, macOS ou Unix :

```
aws docdb describe-pending-maintenance-actions \
   --filters Name=db-cluster-id,Values=sample-cluster1,sample-cluster2
```

Pour Windows :

```
aws docdb describe-pending-maintenance-actions ^
   --filters Name=db-cluster-id,Values=sample-cluster1,sample-cluster2
```

Appliquer les dates

Chaque action de maintenance a une date d'application que vous pouvez trouver lors de la description des actions de maintenance en attente. Lorsque vous lisez le résultat des actions de maintenance en attente depuis le AWS CLI, trois dates sont répertoriées :

- **`CurrentApplyDate`**: date à laquelle l'action de maintenance sera appliquée soit immédiatement, soit lors de la fenêtre de maintenance suivante. Si la maintenance est facultative, cette valeur peut être `null`.
- **`ForcedApplyDate`**—La date à laquelle la maintenance sera automatiquement appliquée, indépendamment de votre fenêtre de maintenance.
- **`AutoAppliedAfterDate`**: date après laquelle la maintenance sera appliquée pendant la fenêtre de maintenance du cluster.

Mises à jour initiées par

En tant qu'utilisateur d'Amazon DocumentDB, vous pouvez initier des mises à jour de vos clusters ou instances. Par exemple, vous pouvez modifier la classe d'une instance en une classe avec plus ou moins de mémoire, ou vous pouvez modifier le groupe de paramètres d'un cluster. Amazon DocumentDB considère ces modifications différemment des mises à jour initiées par Amazon DocumentDB. Pour plus d'informations sur la modification d'un cluster ou d'une instance, consultez ce qui suit :

- Modification d'un cluster Amazon DocumentDB
- Modification d'une instance Amazon DocumentDB

Pour voir une liste des modifications en attente lancées par l'utilisateur, exécutez la commande suivante.

Example

Pour voir les modifications en attente lancées par l'utilisateur pour vos instances

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-instances \
    --query 'DBInstances[*].
[DBClusterIdentifier,DBInstanceIdentifier,PendingModifiedValues]'
```

Pour Windows :

```
aws docdb describe-db-instances ^
    --query 'DBInstances[*].
[DBClusterIdentifier,DBInstanceIdentifier,PendingModifiedValues]'
```

La sortie de cette opération ressemble à ceci (format JSON).

Dans ce cas, `sample-cluster-instance` a un changement en attente vers une classe d'instance `db.r5.xlarge`, alors que `sample-cluster-instance-2` n'a pas de changements en attente.

```
[
    [
        "sample-cluster",
```

```
        "sample-cluster-instance",
        {
            "DBInstanceClass": "db.r5.xlarge"
        }
    ],
    [
        "sample-cluster",
        "sample-cluster-instance-2",
        {}
    ]
]
```

Gestion de vos fenêtres de maintenance Amazon DocumentDB

Chaque instance et chaque cluster est associé à un créneau de maintenance hebdomadaire au cours duquel toutes les modifications en attente sont appliquées. La fenêtre de maintenance est une occasion de contrôler le moment où les modifications et les correctifs logiciels ont lieu, qu'ils aient fait l'objet d'une demande ou qu'ils soient obligatoires. Si un événement de maintenance est planifié pour une semaine donnée, il est déclenché pendant le créneau de maintenance de 30 minutes que vous identifiez. La plupart des événements de maintenance se terminent également au cours du créneau de maintenance de 30 minutes, mais des événements de maintenance plus importants peuvent prendre plus de 30 minutes.

Ce créneau de maintenance de 30 minutes est sélectionné de manière aléatoire sur un bloc horaire de 8 heures par région. Si vous ne spécifiez pas de fenêtre de maintenance préférée lorsque vous créez l'instance ou le cluster, Amazon DocumentDB attribue une fenêtre de maintenance de 30 minutes un jour de la semaine sélectionné au hasard.

Le tableau suivant répertorie pour les différentes régions les blocs de temps à partir desquels les créneaux de maintenance par défaut sont alloués.

Nom de la région	Région	Bloc horaire UTC
USA Est (Ohio)	us-east-2	3 h 00-11 h 00
US East (Virginie du Nord)	us-east-1	3 h 00-11 h 00
USA Ouest (Oregon)	us-west-2	6 h 00-14 h 00
Asie-Pacifique (Hong Kong)	ap-east-1	6 h 00-14 h 00

Nom de la région	Région	Bloc horaire UTC
Asie-Pacifique (Hyderabad)	ap-south-2	6 H 30 — 14 H 30
Asie-Pacifique (Mumbai)	ap-south-1	6 h 00-14 h 00
Asie-Pacifique (Séoul)	ap-northeast-2	13 h 00-21 h 00
Asie-Pacifique (Singapour)	ap-southeast-1	14h00-22h00
Asie-Pacifique (Sydney)	ap-southeast-2	12h00-20h00
Asie-Pacifique (Tokyo)	ap-northeast-1	13 h 00-21 h 00
Canada (Centre)	ca-central-1	3 h 00-11 h 00
Chine (Beijing)	cn-north-1	6 h 00-14 h 00
Chine (Ningxia)	cn-northwest-1	6 h 00-14 h 00
Europe (Francfort)	eu-central-1	21 h 00 - 5 h 00
Europe (Irlande)	eu-west-1	22 h 00-6 h 00
Europe (Londres)	eu-west-2	22 h 00-6 h 00
Europe (Milan)	eu-south-1	02:00-10:00
Europe (Paris)	eu-west-3	23:59-07:29
Moyen-Orient (EAU)	me-central-1	05 H 00 — 13 H 00
Amérique du Sud (São Paulo)	sa-east-1	00:00-08:00
AWS GovCloud (USA Est)	us-gov-east-1	17:00-01:00
AWS GovCloud (US-Ouest)	us-gov-west-1	6 h 00-14 h 00

Modification de vos fenêtres de maintenance Amazon DocumentDB

Le créneau de maintenance doit intervenir au moment où l'utilisation est la plus faible, il peut donc s'avérer nécessaire de le modifier de temps en temps. Votre cluster ou instance est indisponible pendant cette période uniquement si des modifications système (telles qu'une opération de stockage évolutif ou un changement de classe d'instance) sont appliquées et nécessitent une interruption de service. Ensuite, il n'est pas disponible uniquement pendant le délai minimum requis pour apporter les modifications nécessaires.

Pour les mises à niveau du moteur de base de données, Amazon DocumentDB utilise la fenêtre de maintenance préférée du cluster et non la fenêtre de maintenance pour les instances individuelles.

Pour modifier la fenêtre de maintenance

- Pour un cluster, veuillez consulter Modification d'un cluster Amazon DocumentDB.
- Pour une instance, veuillez consulter Modification d'une instance Amazon DocumentDB.

Utilisation des mises à jour du système d'exploitation

Les instances des clusters Amazon DocumentDB nécessitent parfois des mises à jour du système d'exploitation. Amazon DocumentDB met à niveau le système d'exploitation vers une version plus récente afin d'améliorer les performances de la base de données et le niveau de sécurité global des clients. Les mises à jour du système d'exploitation ne modifient pas la version du moteur de cluster ni la classe d'instance d'une instance Amazon DocumentDB.

Nous vous recommandons de mettre à jour d'abord les instances du lecteur dans un cluster, puis l'instance du rédacteur afin d'optimiser la disponibilité de votre cluster. Nous ne recommandons pas de mettre à jour les instances du lecteur et du rédacteur en même temps, car vous risquez de subir des temps d'arrêt plus longs en cas de basculement.

Les mises à jour du système d'exploitation n'ont pas de date d'application et peuvent être appliquées à tout moment. Nous vous recommandons de les appliquer régulièrement pour maintenir vos bases de données Amazon DocumentDB à jour. Amazon DocumentDB n'applique pas ces mises à jour automatiquement. Pour être averti de la disponibilité d'une nouvelle mise à jour facultative, vous pouvez vous inscrire à RDS-EVENT-0230 dans la catégorie des événements d'application de correctifs de sécurité. Pour plus d'informations sur l'abonnement aux événements Amazon DocumentDB, consultez la section Abonnement aux événements Amazon DocumentDB.

Si votre instance est une instance principale, attendez-vous à un basculement de cette dernière lors d'une maintenance sur votre cluster ou votre instance. Pour améliorer votre disponibilité, nous vous recommandons d'utiliser plusieurs instances pour vos clusters Amazon DocumentDB. Pour plus d'informations, consultez Basculement Amazon DocumentDB.

> **Note**
>
> Pour certaines fonctionnalités de gestion, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon Relational Database Service (Amazon RDS).

> **Important**
>
> Votre instance Amazon DocumentDB sera mise hors ligne lors de la mise à niveau du système d'exploitation.

> **Note**
>
> Vous devrez peut-être appliquer toutes les mises à jour facultatives et obligatoires afin de respecter diverses obligations de conformité. Nous vous recommandons d'appliquer régulièrement toutes les mises à jour mises à disposition par Amazon DocumentDB pendant vos fenêtres de maintenance.

Vous pouvez utiliser le AWS Management Console ou le AWS CLI pour déterminer si une mise à jour est facultative ou obligatoire.

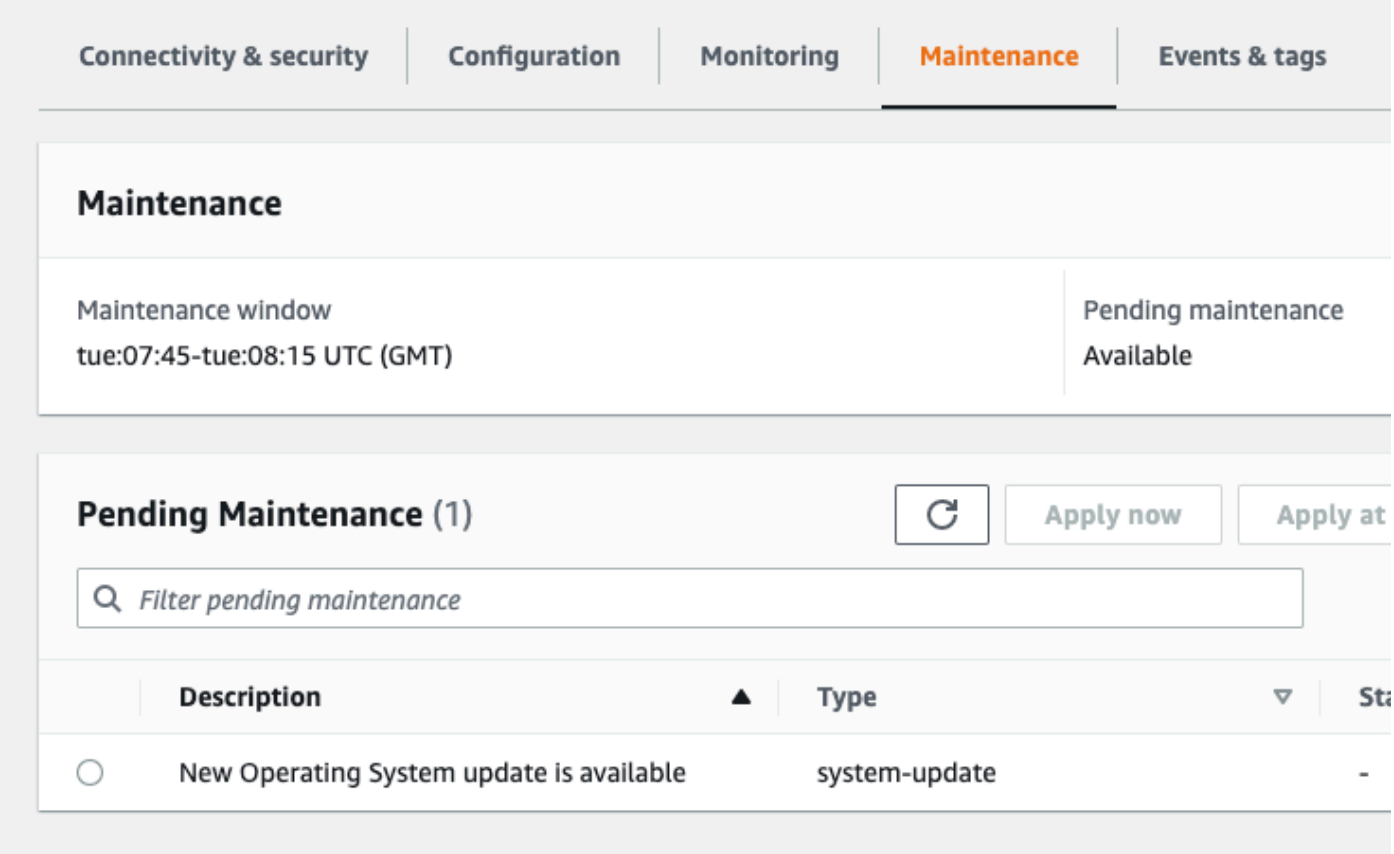
Using the AWS Management Console

Pour déterminer si une mise à jour est facultative ou obligatoire à l'aide du AWS Management Console :

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le volet de navigation, choisissez Clusters, puis sélectionnez l'instance.
3. Choisissez Maintenance.

4. Dans la section Maintenance en attente, recherchez la mise à jour du système d'exploitation et vérifiez la valeur du statut.

Dans le AWS Management Console, l'état de maintenance d'une mise à jour du système d'exploitation est défini sur disponible et n'a pas de date d'application, comme le montre l'image suivante :



Vous pouvez sélectionner la mise à jour du système d'exploitation et cliquer sur Appliquer maintenant ou sur Appliquer à la fenêtre de maintenance suivante dans la section Maintenance en attente. Si la valeur de maintenance correspond à la fenêtre suivante, reportez les éléments de maintenance en choisissant Différer la mise à niveau. Vous ne pouvez pas reporter une action de maintenance en cours.

Vous pouvez également choisir l'instance dans une liste de clusters en cliquant sur Clusters dans le volet de navigation et en sélectionnant Appliquer maintenant ou Appliquer à la prochaine fenêtre de maintenance dans le menu Actions.

Using the AWS CLI

Pour déterminer si une mise à jour est facultative ou obligatoire à l'aide du AWS CLI, appelez la describe-pending-maintenance-actions commande suivante :

```
aws docdb describe-pending-maintenance-actions
```

Une mise à jour obligatoire du système d'exploitation inclut les valeurs `AutoAppliedAfterDate` et `CurrentApplyDate`. Une mise à jour facultative du système d'exploitation n'inclut pas ces valeurs.

Le résultat suivant indique une mise à jour obligatoire du système d'exploitation :

```
{
  "ResourceIdentifier": "arn:aws:docdb:us-east-1:123456789012:db:mydb1",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "AutoAppliedAfterDate": "2022-08-31T00:00:00+00:00",
      "CurrentApplyDate": "2022-08-31T00:00:00+00:00",
      "Description": "New Operating System update is available"
    }
  ]
}
The following output shows an optional operating system update.
{
  "ResourceIdentifier": "arn:aws:docdb:us-east-1:123456789012:db:mydb2",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "Description": "New Operating System update is available"
    }
  ]
}
```

Disponibilité des mises à jour du système d'exploitation

Les mises à jour du système d'exploitation sont spécifiques aux versions du moteur Amazon DocumentDB et aux classes d'instances. Par conséquent, les instances Amazon DocumentDB reçoivent ou nécessitent des mises à jour à des moments différents. Lorsqu'une mise à jour du

système d'exploitation est disponible pour votre instance en fonction de la version du moteur et de la classe d'instance, la mise à jour apparaît dans la console. Il peut également être consulté en exécutant la AWS CLI `describe-pending-maintenance-actions` commande ou en appelant l'opération `DescribePendingMaintenanceActions` API. Si une mise à jour est disponible pour votre instance, vous pouvez mettre à jour votre système d'exploitation en suivant les instructions de la section Appliquer les mises à jour d'Amazon DocumentDB.

Présentation des rôles liés à un service

Amazon DocumentDB (compatible avec MongoDB) utilise des rôles liés à des AWS Identity and Access Management services (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié à Amazon DocumentDB. Les rôles liés au service sont prédéfinis par Amazon DocumentDB et incluent toutes les autorisations dont le service a besoin pour appeler d'autres AWS services en votre nom.

Un rôle lié à un service facilite l'utilisation d'Amazon DocumentDB car vous n'avez pas à ajouter manuellement les autorisations nécessaires. Amazon DocumentDB définit les autorisations de ses rôles liés à un service et, sauf définition contraire, seul Amazon DocumentDB peut assumer ses rôles. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous pouvez supprimer les rôles uniquement après la suppression préalable de leurs ressources connexes. Cela protège vos ressources Amazon DocumentDB car vous ne pouvez pas supprimer par inadvertance l'autorisation d'accès aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés aux services, consultez Services AWS fonctionnant avec IAM et recherchez les services où Oui figure dans la colonne Rôle lié à un service. Choisissez un Yes (oui) ayant un lien permettant de consulter les détails du rôle pour ce service.

Autorisations relatives aux rôles liés au service Amazon DocumentDB

Amazon DocumentDB (compatible avec MongoDB) utilise le rôle lié au service nommé AWSServiceRoleForRDSpour permettre à Amazon DocumentDB d'appeler des AWS services pour le compte de vos clusters.

Le rôle lié à un service AWSServiceRoleForRDS approuve les services suivants pour endosser le rôle :

- `docdb.amazonaws.com`

La politique d'autorisation des rôles permet à Amazon DocumentDB d'effectuer les actions suivantes sur les ressources spécifiées :

- Actions sur `ec2` :
  - `AssignPrivateIpAddresses`
  - `AuthorizeSecurityGroupIngress`
  - `CreateNetworkInterface`
  - `CreateSecurityGroup`
  - `DeleteNetworkInterface`
  - `DeleteSecurityGroup`
  - `DescribeAvailabilityZones`
  - `DescribeInternetGateways`
  - `DescribeSecurityGroups`
  - `DescribeSubnets`
  - `DescribeVpcAttribute`
  - `DescribeVpcs`
  - `ModifyNetworkInterfaceAttribute`
  - `RevokeSecurityGroupIngress`
  - `UnassignPrivateIpAddresses`
- Actions sur `sns` :
  - `ListTopic`
  - `Publish`
- Actions sur `cloudwatch` :
  - `PutMetricData`
  - `GetMetricData`
  - `CreateLogStream`
  - `PullLogEvents`
  - `DescribeLogStreams`
  - `CreateLogGroup`

> **Note**
>
> 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. Il se peut que vous rencontriez le message d'erreur suivant :
> Impossible de créer la ressource. Vérifiez que vous détenez l'autorisation de créer un rôle lié au service. Dans le cas contraire, attendez et réessayez ultérieurement.
> Si vous voyez cette erreur, vérifiez que vous avez respecté les autorisations activées suivantes :
>
> ```
> {
>     "Action": "iam:CreateServiceLinkedRole",
>     "Effect": "Allow",
>     "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
> AWSServiceRoleForRDS",
>     "Condition": {
>         "StringLike": {
>             "iam:AWSServiceName":"rds.amazonaws.com"
>         }
>     }
> }
> ```

Pour plus d'informations, consultez Service-Linked Role Permissions (autorisations du rôle lié à un service) dans le IAM User Guide (guide de l'utilisateur IAM).

Création d'un rôle lié au service Amazon DocumentDB

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous créez un cluster, Amazon DocumentDB crée le rôle lié au service pour vous.

Si vous supprimez ce rôle lié à un service et que vous devez ensuite le recréer, vous pouvez utiliser la même procédure pour recréer le rôle dans votre compte. Lorsque vous créez un cluster, Amazon DocumentDB crée à nouveau le rôle lié au service pour vous.

Modification d'un rôle lié au service Amazon DocumentDB

Amazon DocumentDB ne vous permet pas de modifier le rôle lié au AWSServiceRoleForRDS service. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas changer le nom du rôle, car plusieurs entités peuvent faire référence à ce rôle. Vous pouvez toutefois modifier la

description du rôle à l'aide d'IAM. Pour plus d'informations, consultez Editing a Service-Linked Role (Modification d'un rôle lié à un service) dans le Guide de l'utilisateur IAM.

Supprimer un rôle lié au service Amazon DocumentDB

Si vous n'avez plus besoin d'utiliser une fonction 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 supprimer tous les clusters avant de pouvoir supprimer le rôle lié à un service.

Nettoyage d'un rôle lié au service Amazon DocumentDB

Avant de pouvoir utiliser IAM pour supprimer un rôle lié à un service, vous devez d'abord vérifier qu'aucune session n'est active pour le rôle et supprimer toutes les ressources utilisées par le rôle.

Pour vérifier si une session est active pour le rôle lié à un service dans la console

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à https://console.aws.amazon.com/iam/ l'adresse.
2. Dans le volet de navigation de la console IAM, choisissez Roles, puis choisissez le nom (et non la case à cocher) du AWSServiceRoleForRDSrôle.
3. Sur la page Récapitulatif du rôle sélectionné, choisissez l'onglet Access Advisor.
4. Dans l'onglet Access Advisor, consultez l'activité récente pour le rôle lié à un service.

> **Note**
>
> Si vous ne savez pas si Amazon DocumentDB utilise le AWSServiceRoleForRDS rôle, vous pouvez essayer de le supprimer. Si le service utilise le rôle, la suppression échoue et vous avez accès aux régions dans lesquelles le rôle est utilisé. Si le rôle est utilisé, vous devez attendre que la session se termine avant de pouvoir le supprimer. Vous ne pouvez pas révoquer la session d'un rôle lié à un service.

Si vous souhaitez supprimer le rôle AWSServiceRoleForRDS, vous devez d'abord supprimer toutes vos instances et clusters. Pour plus d'informations sur la suppression des instances et des clusters, consultez les rubriques suivantes :

- Supprimer une instance Amazon DocumentDB

- Suppression d'un cluster Amazon DocumentDB

Régions prises en charge pour les rôles liés au service Amazon DocumentDB

Amazon DocumentDB prend en charge l'utilisation de rôles liés à un service dans toutes les régions où le service est disponible. Pour plus d'informations, veuillez consulter https://docs.aws.amazon.com/documentdb/latest/developerguide/regions-and-azs.html#regions-and-azs-availability.

Utilisation des clusters élastiques Amazon DocumentDB

Les clusters élastiques Amazon DocumentDB prennent en charge des charges de travail comportant des millions de lectures/écritures par seconde et une capacité de stockage de plusieurs pétaoctets. Les clusters élastiques simplifient également la manière dont les développeurs interagissent avec Amazon DocumentDB en éliminant le besoin de choisir, de gérer ou de mettre à niveau des instances.

Les clusters élastiques Amazon DocumentDB ont été créés pour :

- Proposez une solution aux clients à la recherche d'une base de données offrant une évolutivité pratiquement illimitée avec de riches fonctionnalités de requête et une compatibilité avec les API MongoDB.
- Offrez aux clients des limites de connexion plus élevées et réduisez les temps d'arrêt liés à l'application de correctifs.
- Continuez à investir dans une architecture cloud native, élastique et de pointe pour les charges de travail JSON.

Rubriques

Cas d'utilisation d'Elastic Cluster

Les bases de données de documents sont utiles pour les charges de travail qui nécessitent un schéma flexible et rapide, et un développement itératif. Par exemple, les cas d'utilisation d'Amazon DocumentDB, consultez. Cas d'utilisation de base de données de documents

Voici quelques exemples de cas d'utilisation pour lesquels les clusters élastiques peuvent apporter des avantages significatifs :

Profils utilisateurs

Les bases de données documentaires étant dotées d'un schéma flexible, elles peuvent stocker des documents présentant des attributs et des valeurs de données différents à grande échelle. Les clusters élastiques constituent une solution pratique aux profils en ligne dans lesquels différents utilisateurs fournissent différents types d'informations. Supposons que vos applications prennent en charge des centaines de millions de profils d'utilisateurs. Vous pouvez utiliser des clusters élastiques pour prendre en charge de telles applications, car ils peuvent être étendus ou étendus pour prendre en charge des millions d'écritures et de lectures sur ces profils utilisateur. Vous pouvez également réduire les heures creuses afin de réduire les coûts.

Gestion du contenu et enregistrements historiques

Pour gérer de manière efficace le contenu, vous devez être en mesure de collecter et regrouper du contenu à partir d'une variété de sources, pour ensuite le délivrer au client. Grâce à leur schéma flexible, les bases de données de documents sont parfaites pour collecter et stocker tous les types de données. Vous pouvez les utiliser pour créer et intégrer de nouveaux types de contenu, notamment du contenu généré par les utilisateurs, tel que des images, des commentaires et des vidéos. Au fil du temps, votre base de données peut avoir besoin d'un espace de stockage supplémentaire. Avec les clusters élastiques, vous pouvez répartir vos données sur un plus grand nombre de volumes de stockage, ce qui vous permet de stocker des pétaoctets de données dans un seul cluster.

Avantages des clusters élastiques

AWS intégration des services

Les clusters élastiques Amazon DocumentDB s'intègrent aux autres AWS services de la même manière qu'Amazon DocumentDB :

- Migration : vous pouvez utiliser AWS Database Migration Service (DMS) pour migrer de MongoDB et d'autres bases de données relationnelles vers des clusters élastiques Amazon DocumentDB.
- Surveillance : vous pouvez surveiller l'état et les performances de votre cluster élastique à l'aide d'Amazon CloudWatch.
- Sécurité - Vous pouvez configurer l'authentification et l'autorisation via AWS Identity and Access Management (IAM) pour gérer vos clusters élastiques et utiliser Amazon VPC pour des connexions sécurisées uniquement VPC.
- Gestion des données : vous pouvez l'utiliser AWS Glue pour importer et exporter des données depuis/vers d'autres AWS services tels qu'Amazon S3, Amazon Redshift et OpenSearch Amazon Service.

Région du cluster élastique et disponibilité des versions

Disponibilité dans les Régions

Le tableau suivant indique les AWS régions dans lesquelles les clusters élastiques Amazon DocumentDB sont actuellement disponibles, ainsi que le point de terminaison de chaque région.

Nom de la région	Région	Zones de disponibilité
USA Est (Virginie du Nord)	`us-east-1`	5
USA Est (Ohio)	`us-east-2`	3
USA Ouest (Oregon)	`us-west-2`	3
Asie-Pacifique (Mumbai)	`ap-south-1`	3
Asie-Pacifique (Séoul)	`ap-northeast-2`	3
Asie-Pacifique (Singapour)	`ap-southeast-1`	3
Asie-Pacifique (Sydney)	`ap-southeast-2`	3
Asie-Pacifique (Tokyo)	`ap-northeast-1`	3
Amérique du Sud (São Paulo)	`sa-east-1`	3

Nom de la région	Région	Zones de disponibilité
Europe (Francfort)	`eu-central-1`	3
Europe (Irlande)	`eu-west-1`	3
Europe (Londres)	`eu-west-2`	3

Disponibilité des versions

Les clusters Elastic prennent en charge le protocole filaire compatible avec MongoDB 5.0. Pour connaître les différences entre les clusters basés sur des instances DocumentDB 4.0 et les clusters élastiques, consultez. Différences fonctionnelles entre Amazon DocumentDB 4.0 et les clusters élastiques

Limites

Gestion élastique des clusters

Les fonctionnalités et fonctionnalités de gestion de cluster suivantes ne sont pas prises en charge dans cette version :

- Possibilité de créer des clusters mondiaux
- Événements Amazon DocumentDB existants et abonnement à des événements
- Sharding de plage
- Partager une collection existante
- Clé partagée à champs multiples
- Modifier la clé de partition
- oint-in-time Restauration du PC
- Clonage
- Performance Insights

> **Note**
>
> Pour plus d'informations sur les limites des clusters élastiques, consultezQuotas et limites Amazon DocumentDB.

Opérations de requête et d'écriture

Les commandes et fonctionnalités d'opération de requête et d'écriture suivantes ne sont pas prises en charge dans cette version :

- Commandes DDL lors des opérations de dimensionnement
- Profiler
- Groupes de paramètres
- AWS Config
- AWS Backup

Gestion des collections et des index

Les fonctionnalités de gestion des collections et des index suivantes ne sont pas prises en charge dans cette version :

- Indices géospatiaux
- Création d'un index d'arrière-plan

Administration et diagnostic

Les commandes et fonctionnalités d'administration et de diagnostic suivantes ne sont pas prises en charge dans cette version :

- AWS Secrets Manager
- Rôles personnalisés R ole-based-access-control (RBAC).
- Lors de la connexion, le problème d'écriture de 0 n'est pas pris en charge.
- Modification des sous-réseaux appartenant à un VPC qui n'est actuellement pas attribué à un cluster élastique existant.

Fonctionnalités d'inscription

Les fonctionnalités opt-in Amazon DocumentDB suivantes ne sont pas prises en charge dans cette version :

- Transactions ACID
- Audit DDL/DML
- Change streams
- Commandes de session

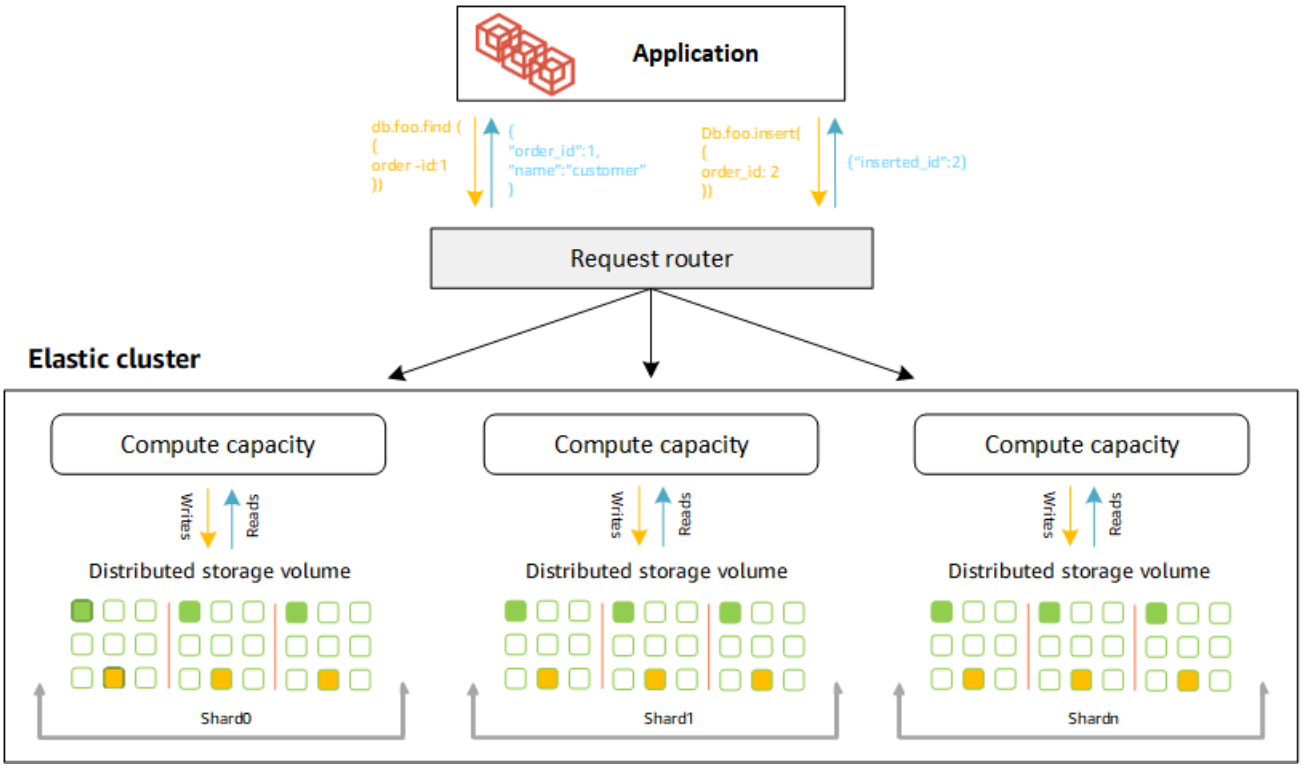
Clusters élastiques Amazon DocumentDB : comment ça marche

Les rubriques de cette section fournissent des informations sur les mécanismes et les fonctions qui alimentent les clusters élastiques Amazon DocumentDB.

Rubriques

Sharding élastique de clusters Amazon DocumentDB

Les clusters élastiques Amazon DocumentDB utilisent le sharding basé sur le hachage pour partitionner les données sur un système de stockage distribué. Le sharding, également appelé partitionnement, divise les grands ensembles de données en petits ensembles de données répartis sur plusieurs nœuds, ce qui vous permet d'étendre votre base de données au-delà des limites de mise à l'échelle verticale. Les clusters élastiques utilisent la séparation, ou « découplage », du calcul et du stockage dans Amazon DocumentDB, ce qui vous permet d'évoluer indépendamment les uns des autres. Plutôt que de repartitionner les collections en déplaçant de petits morceaux de données entre les nœuds de calcul, les clusters élastiques copient les données de manière efficace dans le système de stockage distribué.



Définitions de partitions

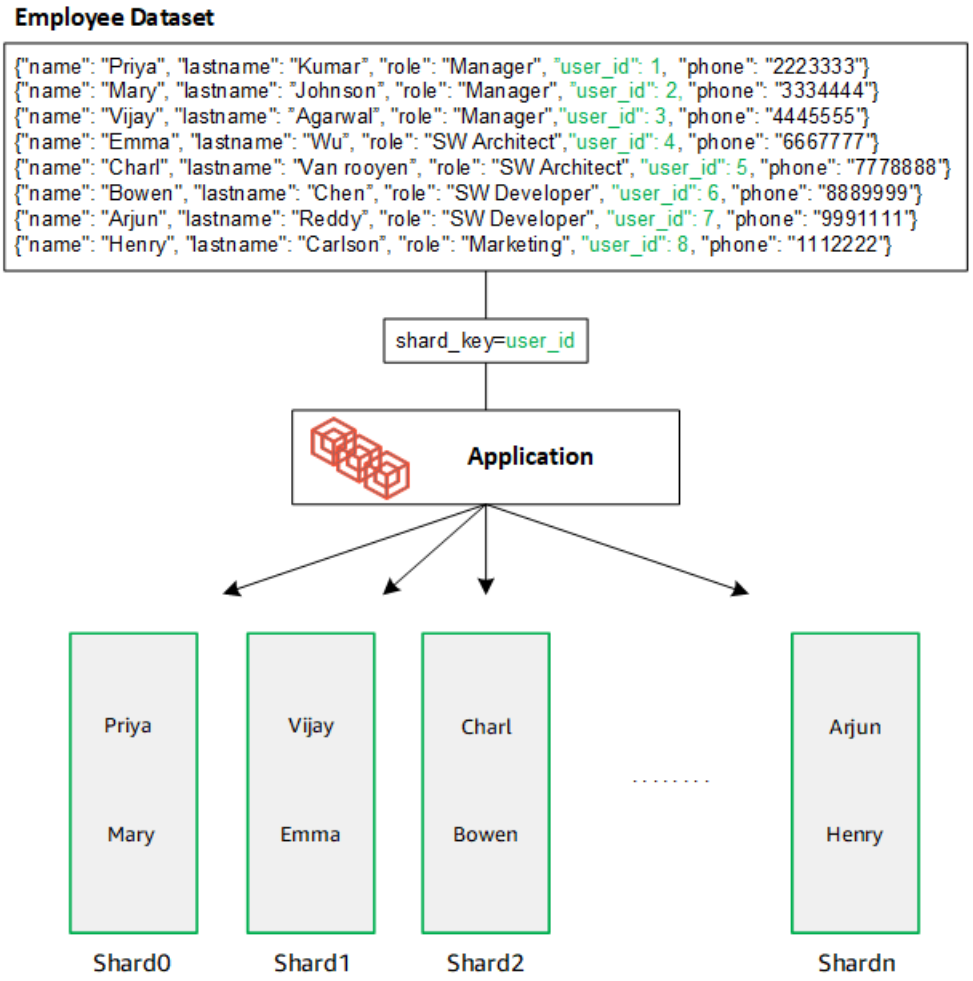
Définitions de la nomenclature des partitions :

- Shard : un shard fournit le calcul nécessaire à un cluster élastique. Par défaut, une partition comportera deux nœuds. Vous pouvez configurer un maximum de 32 partitions et chaque partition peut comporter un maximum de 64 vCPU.
- Clé de partition : une clé de partition est un champ obligatoire dans vos documents JSON dans les collections fragmentées que les clusters élastiques utilisent pour distribuer le trafic de lecture et d'écriture vers le fragment correspondant.
- Collection de partitions : une collection de partitions est une collection dont les données sont réparties sur un cluster élastique sous forme de partitions de données.
- Partition : une partition est une portion logique de données fragmentées. Lorsque vous créez une collection fragmentée, les données sont automatiquement organisées en partitions au sein de chaque partition en fonction de la clé de partition. Chaque partition possède plusieurs partitions.

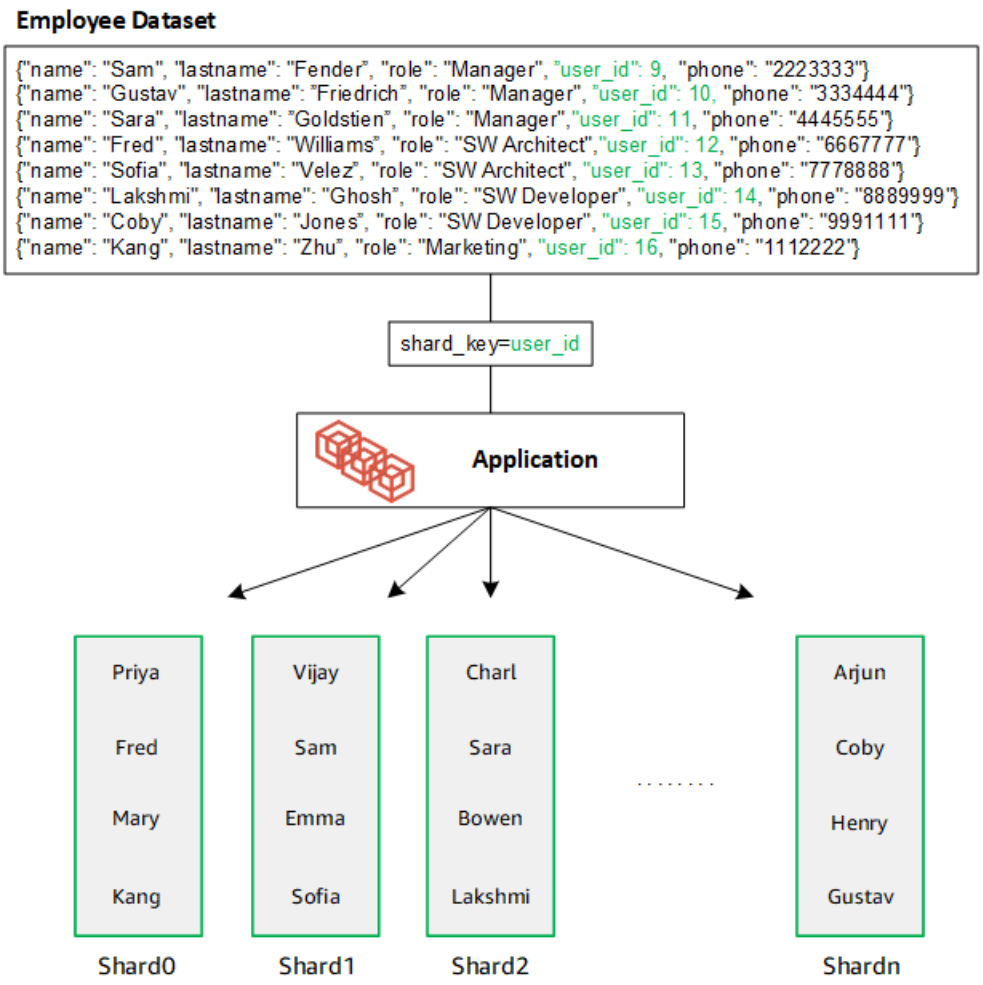
Répartition des données entre des partitions configurées

Créez une clé de partition dotée de nombreuses valeurs uniques. Une bonne clé de partition partitionnera uniformément vos données entre les partitions sous-jacentes, offrant ainsi à votre

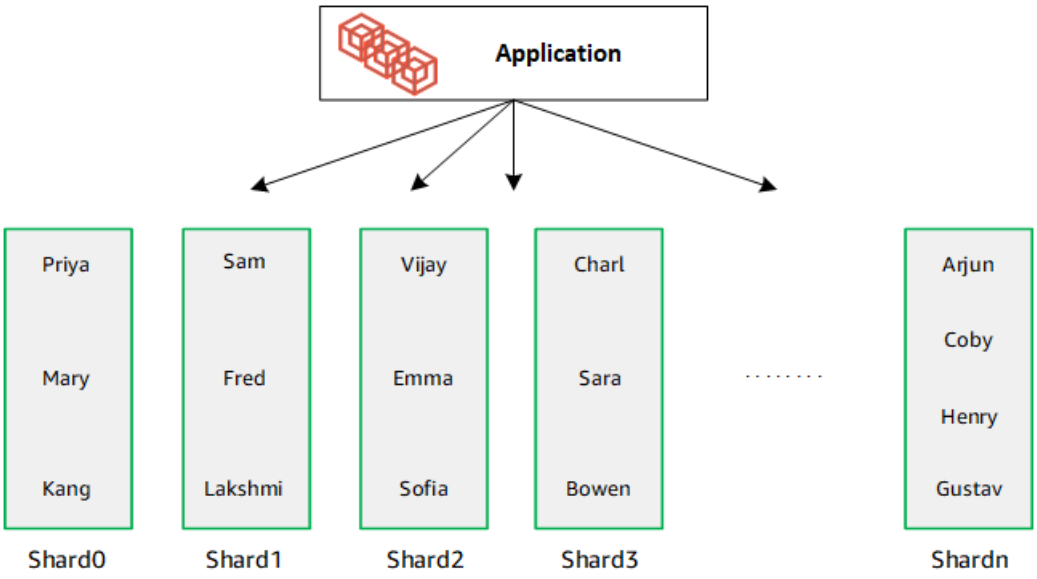
charge de travail le meilleur débit et les meilleures performances. L'exemple suivant concerne les données relatives aux noms d'employés qui utilisent une clé de partition nommée « user_id » :



DocumentDB utilise le hachage pour partitionner vos données entre les partitions sous-jacentes. Les données supplémentaires sont insérées et distribuées de la même manière :



Lorsque vous agrandissez votre base de données en ajoutant des partitions supplémentaires, Amazon DocumentDB redistribue automatiquement les données :



Migration élastique de clusters

Amazon DocumentDB prend en charge la migration des données partitionnées MongoDB vers des clusters élastiques. Les méthodes de migration hors ligne, en ligne et hybrides sont prises en charge. Pour plus d'informations, consultez Migration vers Amazon DocumentDB.

Mise à l'échelle élastique des clusters

Amazon DocumentDB Elastic Clusters permet d'augmenter le nombre de partitions (scalage externe) dans votre cluster élastique, ainsi que le nombre de vCPU appliqués à chaque partition (scalabilité). Vous pouvez également réduire le nombre de partitions et la capacité de calcul (vCPU) selon les besoins.

Pour connaître les meilleures pratiques en matière de dimensionnement, voirMise à l'échelle des clusters élastiques.

> **Note**
>
> La mise à l'échelle au niveau du cluster est également disponible. Pour plus d'informations, consultez Dimensionnement des clusters Amazon DocumentDB.

Fiabilité du cluster élastique

Amazon DocumentDB est conçu pour être fiable, durable et tolérant aux pannes. Pour améliorer la disponibilité, Elastic Clusters déploie deux nœuds par partition placés dans différentes zones de disponibilité. Amazon DocumentDB inclut plusieurs fonctionnalités automatiques qui en font une solution de base de données fiable. Pour plus d'informations, consultez Fiabilité d'Amazon DocumentDB.

Stockage et disponibilité élastiques en cluster

Les données Amazon DocumentDB sont stockées dans un volume de cluster, qui est un volume virtuel unique qui utilise des disques SSD (Solid State Drive). Un volume de cluster se compose de six copies de vos données, qui sont répliquées automatiquement sur plusieurs zones de disponibilité au sein d'une même AWS région. Cette réplication garantit que vos données sont hautement durables, avec une possibilité moindre de perte des données. Elle permet également de vous assurer que votre cluster est plus disponible pendant un basculement, car les copies de vos données existent

déjà dans d'autres zones de disponibilité. Pour plus de détails sur le stockage, la haute disponibilité et la réplication, consultezAmazon DocumentDB : comment cela fonctionne.

Différences fonctionnelles entre Amazon DocumentDB 4.0 et les clusters élastiques

Les différences fonctionnelles suivantes existent entre Amazon DocumentDB 4.0 et les clusters élastiques.

- Les résultats provenant de `top` et `collStats` sont partitionnés par fragments. Pour les collections fragmentées, les données sont réparties entre plusieurs partitions et les `collStats` rapports sont agrégés `collScans` à partir des partitions.
- Les statistiques de collecte provenant de `top` et `collStats` pour les collections fragmentées sont réinitialisées lorsque le nombre de partitions du cluster est modifié.
- Le rôle intégré de sauvegarde est désormais compatible`serverStatus`. Action : les développeurs et les applications dotés d'un rôle de sauvegarde peuvent collecter des statistiques sur l'état du cluster Amazon DocumentDB.
- Le `SecondaryDelaySecs` champ est remplacé `slaveDelay` dans la `replSetGetConfig` sortie.
- La `hello` commande remplace `isMaster` - `hello` renvoie un document qui décrit le rôle du cluster élastique.
- Dans les clusters élastiques, `$elemMatch` l'opérateur correspond uniquement aux documents du premier niveau d'imbrication d'un tableau. Dans Amazon DocumentDB 4.0, l'opérateur parcourt tous les niveaux avant de renvoyer les documents correspondants. Par exemple :

```
db.foo.insert(
[
    {a: {b: 5}},
    {a: {b: [5]}},
    {a: {b: [3, 7]}},
    {a: [{b: 5}]},
    {a: [{b: 3}, {b: 7}]},
    {a: [{b: [5]}]},
    {a: [{b: [3, 7]}]},
    {a: [[{b: 5}]]},
    {a: [[{b: 3}, {b: 7}]]},
```

```
    {a: [[{b: [5]}]]},
    {a: [[{b: [3, 7]}]]}
]);
// Elastic Clusters
> db.foo.find({a: {$elemMatch: {b: {$elemMatch: {$lt: 6, $gt: 4}}}}}, {_id: 0})
{ "a" : [ { "b" : [ 5 ] } ] }

// Docdb 4.0: traverse more than one level deep
> db.foo.find({a: {$elemMatch: {b: {$elemMatch: {$lt: 6, $gt: 4}}}}}, {_id: 0})
{ "a" : [ { "b" : [ 5 ] } ] }
{ "a" : [ [ { "b" : [ 5 ] } ] ] }
```

- La projection « $ » dans Amazon DocumentDB 4.0 renvoie tous les documents avec tous les champs. Avec les clusters élastiques, la `find` commande avec une projection « $ » renvoie les documents qui correspondent au paramètre de requête contenant uniquement le champ correspondant à la projection « $ ».
- Dans les clusters élastiques, `find` les commandes avec les paramètres de `$options` requête `$regex` et de requête renvoient une erreur : « Impossible de définir les options à la fois dans $regex et $options ».

- Avec les clusters élastiques, renvoie `$indexOfCP` désormais « -1 » lorsque :
  - la sous-chaîne est introuvable dans le`string expression`, ou
  - `start`est un nombre supérieur à`end`, ou
  - `start`est un nombre supérieur à la longueur en octets de la chaîne.

  Dans Amazon DocumentDB 4.0, `$indexOfCP` renvoie « 0 » lorsque la `start` position est supérieure à un nombre `end` ou à la longueur en octets de la chaîne.
- Avec les clusters élastiques, les opérations de projection dans`_id fields`, par exemple :`{"_id.nestedField" : 1}`, renvoient des documents qui incluent uniquement le champ projeté. Alors que dans Amazon DocumentDB 4.0, les commandes de projection de champs imbriqués ne filtrent aucun document.

Démarrez avec les clusters élastiques Amazon DocumentDB

Cette section de mise en route explique comment créer et interroger votre premier cluster élastique. Il existe de nombreuses manières de se connecter et de démarrer avec des clusters élastiques. Ce

guide utilise AWS Cloud9un terminal Web pour connecter et interroger votre cluster élastique à l'aide du shell mongo directement depuis le AWS Management Console.

Rubriques

Configuration

Si vous préférez vous connecter à votre Amazon DocumentDB depuis votre machine locale en créant une connexion SSH vers une instance Amazon EC2, consultez Connexion à Amazon EC2.

Prérequis

Avant de créer votre premier cluster Amazon DocumentDB, vous devez effectuer les opérations suivantes :

Créez un compte Amazon Web Services (AWS)

Avant de pouvoir commencer à utiliser Amazon DocumentDB, vous devez disposer d'un compte Amazon Web Services (AWS). Le AWS compte est gratuit. Vous payez uniquement les services et les ressources que vous utilisez.

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

Pour vous inscrire à un Compte AWS

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

   Dans le cadre de la procédure d‘inscription, vous recevrez un appel téléphonique 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. Pour des raisons de sécurité, attribuez un accès administratif à un utilisateur et utilisez uniquement l'utilisateur root pour effectuer les tâches nécessitant un accès utilisateur root.

Configurez les autorisations AWS Identity and Access Management (IAM) nécessaires.

L'accès à la gestion des ressources Amazon DocumentDB telles que les clusters, les instances et les groupes de paramètres de cluster nécessite des informations d'identification AWS pouvant être utilisées pour authentifier vos demandes. Pour plus d'informations, consultez Identity and Access Management pour Amazon DocumentDB.

1. Dans la barre de recherche du AWS Management Console, tapez IAM et sélectionnez IAM dans le menu déroulant.
2. Une fois dans la console IAM, sélectionnez Utilisateurs dans le volet de navigation.
3. Sélectionnez votre nom d'utilisateur.
4. Cliquez sur le bouton Ajouter des autorisations.
5. Sélectionnez Attach existing policies directly (Attacher directement les politiques existantes).
6. Tapez `AmazonDocDBFullAccess` dans la barre de recherche et sélectionnez-la une fois qu'elle apparaît dans les résultats de recherche.
7. Cliquez sur le bouton bleu en bas qui indique Suivant : Réviser.
8. Cliquez sur le bouton bleu en bas qui indique Ajouter des autorisations.

Création d'un Amazon Virtual Private Cloud (Amazon VPC)

Cette étape n'est nécessaire que si vous ne possédez pas encore d'Amazon VPC par défaut. Si ce n'est pas le cas, suivez l'étape 1 de la section Getting Started with Amazon VPC User Guide. Cela prendra moins de cinq minutes.

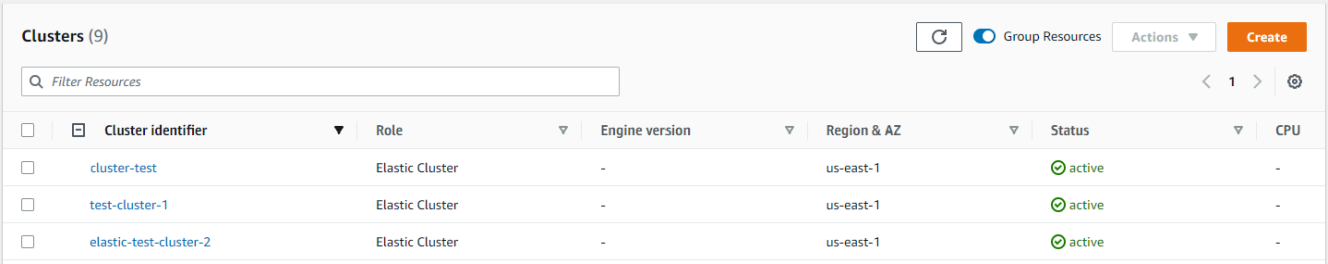
Étape 1 : Création d'un cluster élastique

Dans cette section, nous expliquons comment créer un tout nouveau cluster élastique, en utilisant AWS Management Console ou AWS CLI en suivant les instructions suivantes.

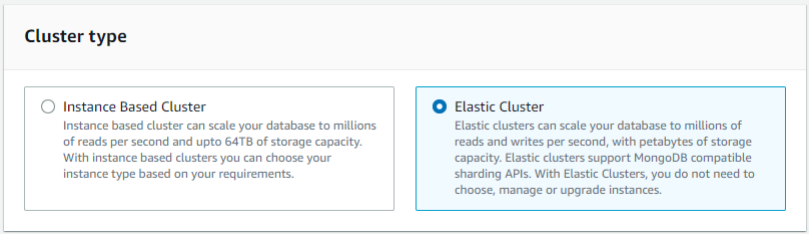
Using the AWS Management Console

Pour créer une configuration de cluster élastique à l'aide de AWS Management Console :

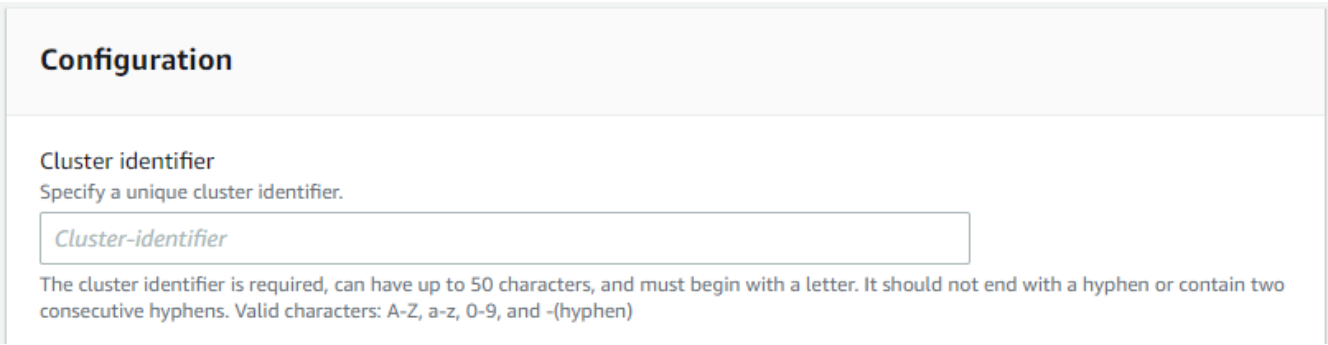
1. Connectez-vous à la console Amazon DocumentDB AWS Management Consoleet ouvrez-la.
2. Sur la console de gestion Amazon DocumentDB, sous Clusters, choisissez Create.



3. Sur la page Créer un cluster Amazon DocumentDB, dans la section Type de cluster, sélectionnez Elastic Cluster.



4. Sur la page Créer un cluster Amazon DocumentDB, dans la section Configuration, entrez un identifiant de cluster unique (conformément aux exigences de dénomination situées sous le champ).



5. Pour les champs de configuration du shard :

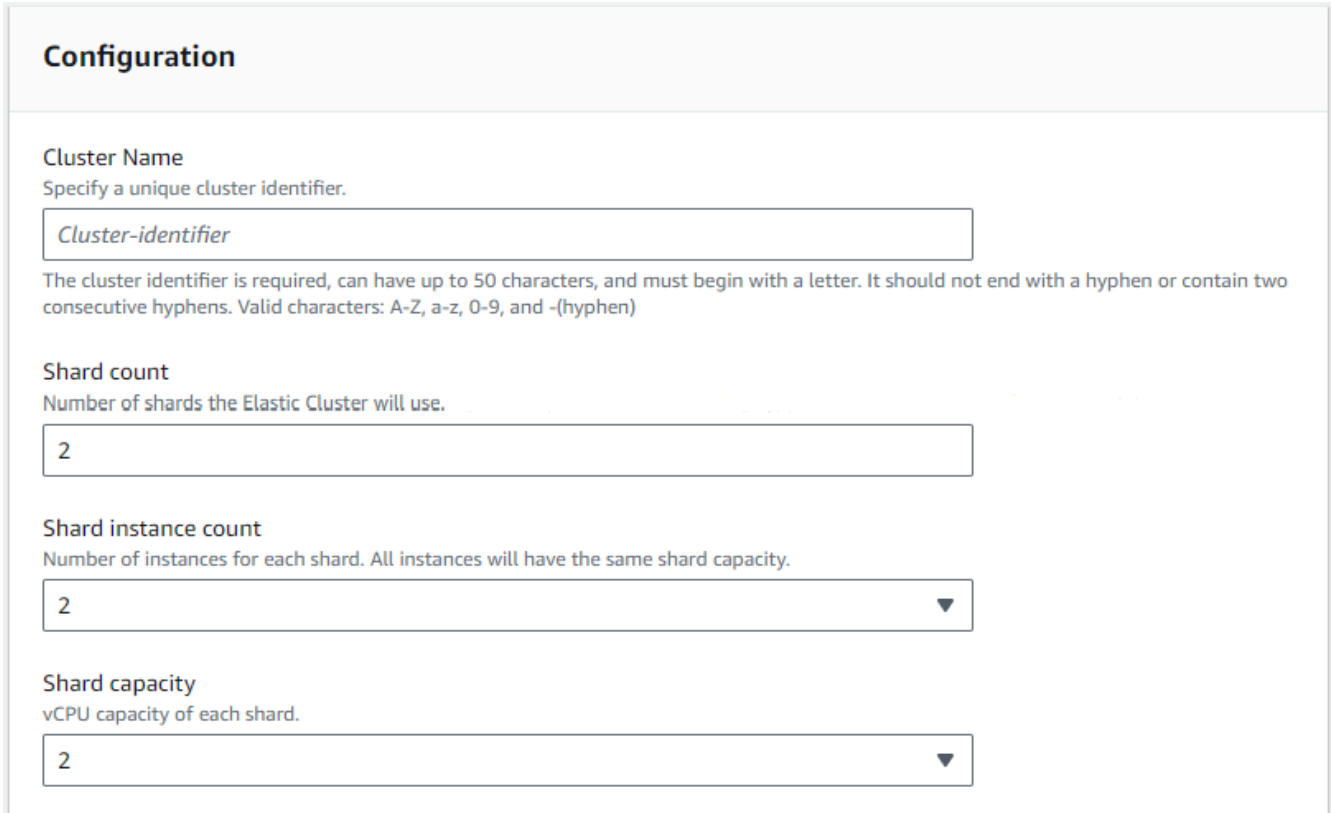
   a. Dans le champ Nombre de partitions, entrez le nombre de partitions que vous souhaitez ajouter à votre cluster. Le nombre maximum de partitions par cluster est de 32.

> **Note**
>
> Deux nœuds seront déployés pour chaque partition. Les deux nœuds auront la même capacité de partition.

b. Dans le champ Nombre d'instances de partition, choisissez le nombre d'instances de répliques que vous souhaitez associer à chaque partition. Le nombre maximum d'instances de partition est de 16, par incréments de 1. Toutes les instances de réplica ont la même capacité de partition que celle définie dans le champ suivant.

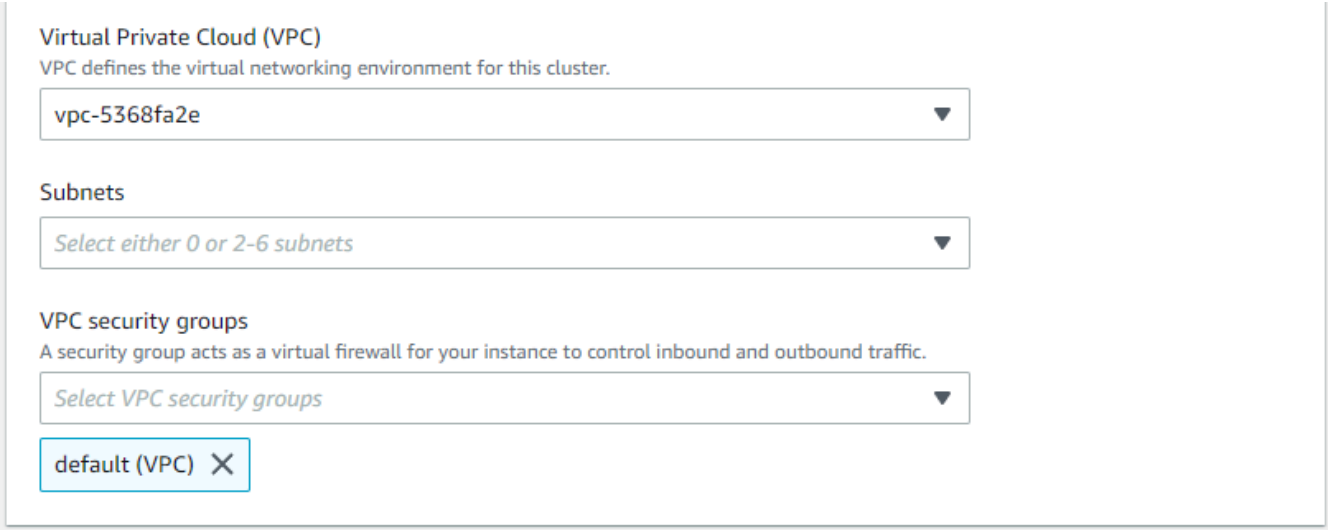
> **Note**
>
> Le nombre d'instances de répliques s'applique à tous les fragments du cluster élastique. Une valeur de 1 pour le nombre d'instances de partition signifie qu'il existe une instance d'écriture et que toutes les instances supplémentaires sont des répliques qui peuvent être utilisées pour les lectures et pour améliorer la disponibilité.

c. Dans le champ Capacité de partition, choisissez le nombre de processeurs virtuels (vCPU) que vous souhaitez associer à chaque instance de partition. Le nombre maximum de vCPU par instance de partition est de 64. Les valeurs autorisées sont 2, 4, 8, 16, 32, 64.



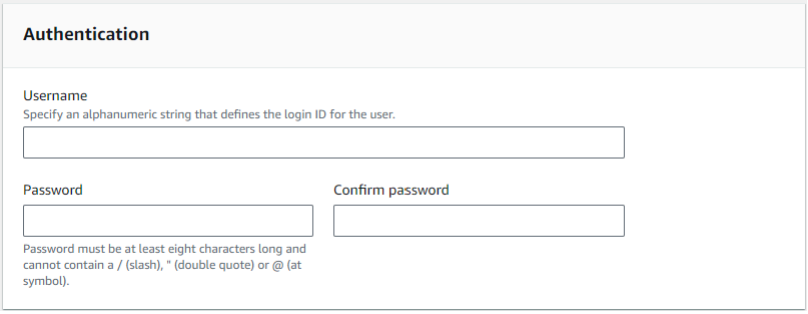
6. Dans le champ Virtual Private Cloud (VPC), sélectionnez un VPC dans la liste déroulante.

Pour les sous-réseaux et les groupes de sécurité VPC, vous pouvez utiliser les valeurs par défaut ou sélectionner trois sous-réseaux de votre choix et jusqu'à trois groupes de sécurité VPC (un minimum).



7. Dans la section Authentification, entrez une chaîne identifiant le nom de connexion de l'utilisateur principal dans le champ Nom d'utilisateur.

   Dans le champ Mot de passe, entrez un mot de passe unique conforme aux instructions.

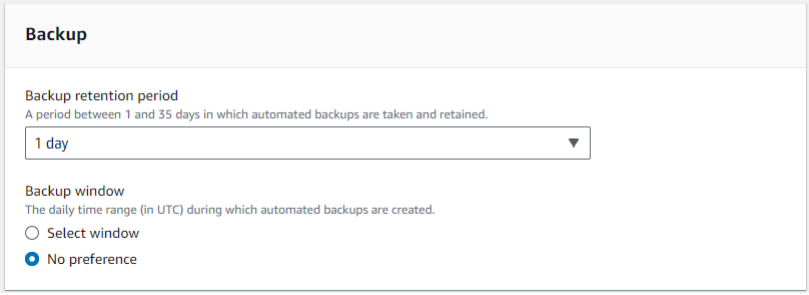


8. Dans la section Chiffrement, conservez les paramètres par défaut.

   Vous pouvez éventuellement saisir un AWS KMS key ARN que vous avez créé. Pour plus d'informations, consultez Chiffrement des données au repos pour les clusters Amazon DocumentDB Elastic Cluster DB..

> **Important**
>
> Le chiffrement doit être activé pour les clusters élastiques.

9. Dans la section Backup, modifiez les champs en fonction de vos besoins en matière de sauvegarde.



   a. Période de conservation des sauvegardes : dans la liste, choisissez le nombre de jours pendant lesquels vous pouvez conserver les sauvegardes automatiques de ce cluster avant de les supprimer.

   b. Fenêtre de sauvegarde : définissez l'heure et la durée quotidiennes pendant lesquelles Amazon DocumentDB doit effectuer des sauvegardes de ce cluster.

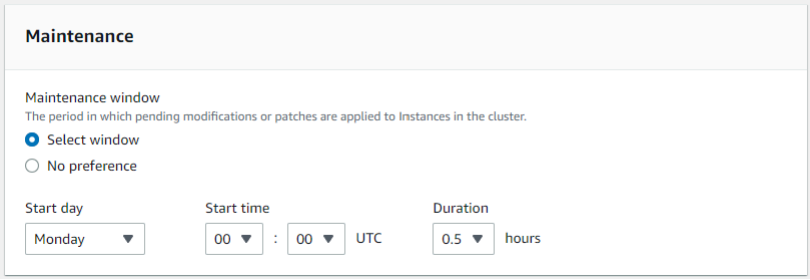
      i. Choisissez Sélectionner une fenêtre si vous souhaitez configurer l'heure et la durée de création des sauvegardes.

         Heure de début : dans la première liste, choisissez l'heure de début (UTC) pour démarrer vos sauvegardes automatiques. Dans la deuxième liste, choisissez la minute de l'heure à laquelle vous voulez que les sauvegardes automatiques commencent.

         Durée : dans la liste, choisissez le nombre d'heures à allouer à la création de sauvegardes automatiques.

      ii. Choisissez Aucune préférence si vous souhaitez qu'Amazon DocumentDB choisisse l'heure et la durée de création des sauvegardes.

10. Dans la section Maintenance, choisissez le jour, l'heure et la durée pendant lesquels les modifications ou les correctifs sont appliqués à votre cluster.



11. Choisissez Créer un cluster.

Le cluster élastique est en cours de provisionnement. Cette opération peut prendre jusqu'à quelques minutes. Vous pouvez vous connecter à votre cluster lorsque l'état du cluster élastique s'affiche, comme **`active`** dans la liste des clusters.

Using the AWS CLI

Pour créer un cluster élastique à l'aide de AWS CLI, utilisez l'`create-cluster`opération avec les paramètres suivants :

- --cluster-name : obligatoire. Nom actuel du cluster d'échelles élastiques tel qu'il a été saisi lors de sa création ou de sa dernière modification.
- --shard-capacity : obligatoire. Le nombre de vCPU assignés à chaque partition. Le maximum est de 64. Les valeurs autorisées sont 2, 4, 8, 16, 32, 64.
- --shard-count : obligatoire. Le nombre de partitions attribuées au cluster. Le maximum est de 32.
- --shard-instance-count—Facultatif. Le nombre d'instances de répliques s'appliquant à toutes les partitions de ce cluster. Le maximum est de 16.
- --admin-user-name : obligatoire. Le nom d'utilisateur associé à l'utilisateur administrateur.
- --admin-user-password : obligatoire. Le mot de passe associé à l'utilisateur administrateur.
- --auth-type : obligatoire. Type d'authentification utilisé pour déterminer où récupérer le mot de passe utilisé pour accéder au cluster élastique. Les types valides sont `PLAIN_TEXT` ou`SECRET_ARN`.
- --vpc-security-group-ids—Facultatif. Configurez une liste de groupes de sécurité VPC EC2 à associer à ce cluster.
- --preferred-maintenance-window—Facultatif. Configurez la plage horaire hebdomadaire pendant laquelle la maintenance du système peut avoir lieu, en temps universel coordonné (UTC).

  Le format est :`ddd:hh24:mi-ddd:hh24:mi`. Jours valides (ddd) : lundi, mardi, mercredi, jeudi, vendredi, samedi, dimanche

  La valeur par défaut est une fenêtre de 30 minutes sélectionnée au hasard sur une période de 8 heures pour chaque région Amazon Web Services, survenant un jour aléatoire de la semaine.

  Fenêtre minimale de 30 minutes.
- --kms-key-id—Facultatif. Configurez l'identifiant de clé KMS pour un cluster chiffré.

L'identifiant de clé KMS est l'Amazon Resource Name (ARN) de la clé de AWS KMS chiffrement. Si vous créez un cluster à l'aide du même compte Amazon Web Services qui possède la clé de chiffrement KMS utilisée pour chiffrer le nouveau cluster, vous pouvez utiliser l'alias de clé KMS au lieu de l'ARN pour la clé de chiffrement KMS.

Si aucune clé de chiffrement n'est spécifiée dans KmsKeyId et si le `StorageEncrypted` paramètre est vrai, Amazon DocumentDB utilise votre clé de chiffrement par défaut.

- --preferred-backup-window—Facultatif. Période quotidienne préférée pendant laquelle les sauvegardes automatisées sont créées. La valeur par défaut est une fenêtre de 30 minutes sélectionnée au hasard dans un intervalle de 8 heures pour chacune d'entre elles. Région AWS
- --backup-retention-period—Facultatif. Nombre de jours de conservation des sauvegardes automatiques. La valeur par défaut est 1.
- --storage-encrypted—Facultatif. Configure si le cluster est chiffré ou non.

  `--no-storage-encrypted`indique que le cluster n'est pas chiffré.
- --subnet-ids—Facultatif. Configurez les identifiants de sous-réseau.

Dans les exemples suivants, remplacez chaque *`espace réservé pour l'entrée utilisateur`* par vos propres informations.

> **Note**
>
> Les exemples suivants incluent la création d'une clé KMS spécifique. Pour utiliser la clé KMS par défaut, n'incluez pas le `--kms-key-id` paramètre.

Pour Linux, macOS ou Unix :

```
aws docdb-elastic create-cluster \
     --cluster-name sample-cluster-123 \
     --shard-capacity 8 \
     --shard-count 4 \
     --shard-instance-count 3 \
     --auth-type PLAIN_TEXT \
     --admin-user-name testadmin \
     --admin-user-password testPassword \
     --vpc-security-group-ids ec-65f40350 \
```

```
     --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 \
     --subnet-ids subnet-9253c6a3, subnet-9f1b5af9 \
     --preferred-backup-window 18:00-18:30 \
     --backup-retention-period 7
```

Pour Windows :

```
aws docdb-elastic create-cluster ^
     --cluster-name sample-cluster-123 ^
     --shard-capacity 8 ^
     --shard-count 4 ^
     --shard-instance-count 3 ^
     --auth-type PLAIN_TEXT ^
     --admin-user-name testadmin ^
     --admin-user-password testPassword ^
     --vpc-security-group-ids ec-65f40350 ^
     --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 ^
     --subnet-ids subnet-9253c6a3, subnet-9f1b5af9 \
     --preferred-backup-window 18:00-18:30 \
     --backup-retention-period 7
```

Étape 2 : Création d'un AWS Cloud9 environnement

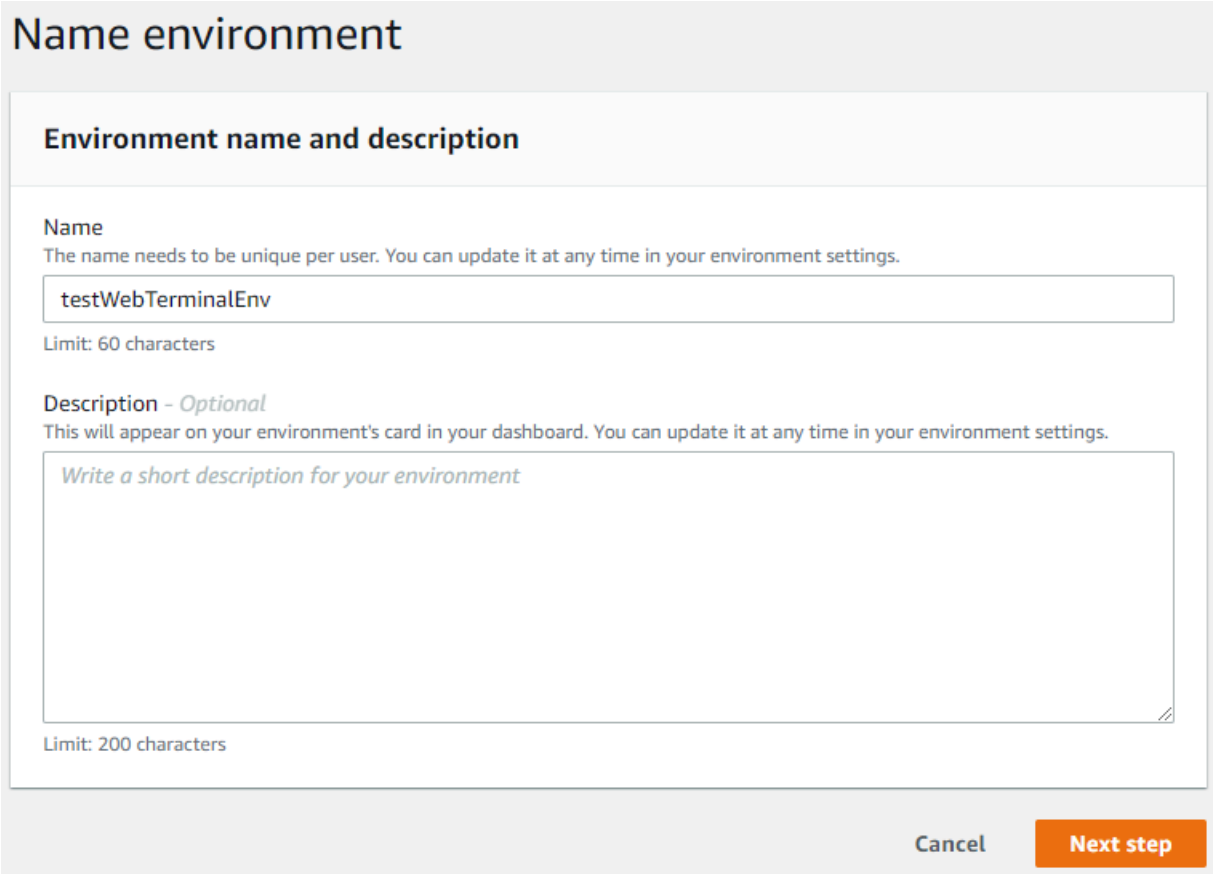
AWS Cloud9 fournit un terminal Web que vous pouvez utiliser pour vous connecter à vos clusters élastiques Amazon DocumentDB et les interroger à l'aide du shell mongo.

> **Note**
>
> Remarque : Votre AWS Cloud9 environnement doit appartenir au même groupe de sécurité que votre instance. Vous pouvez modifier le groupe de sécurité dans la console Amazon EC2.

1. Utilisez votre AWS compte et accédez au AWS Management Console.
2. Accédez à la AWS Cloud9 console. Vous pouvez taper « Cloud9 » dans le champ de recherche pour le localiser.
3. Sur la page d'accueil de AWS Cloud9l'environnement, choisissez Créer un environnement.

4. Sur la page Nom de l'environnement, dans le champ Nom, entrez le nom de votre choix.

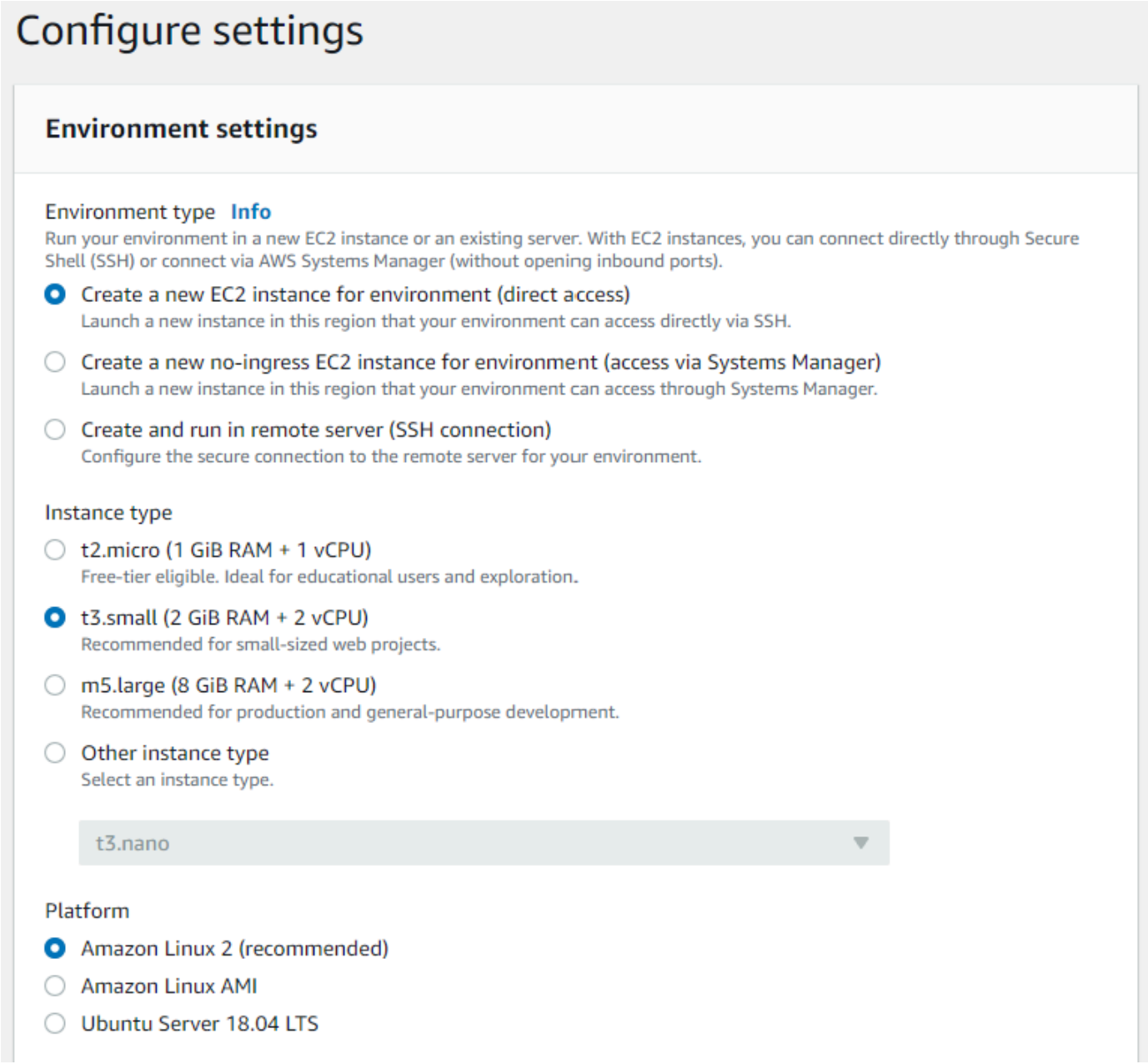
   Choisissez Next step (Étape suivante).



5. Dans Paramètres d'environnement, dans la section Type d'environnement, sélectionnez Créer une nouvelle instance EC2 pour l'environnement (accès direct).

   Dans la section Type d'instance, sélectionnez un type d'instance approprié pour votre réseau.

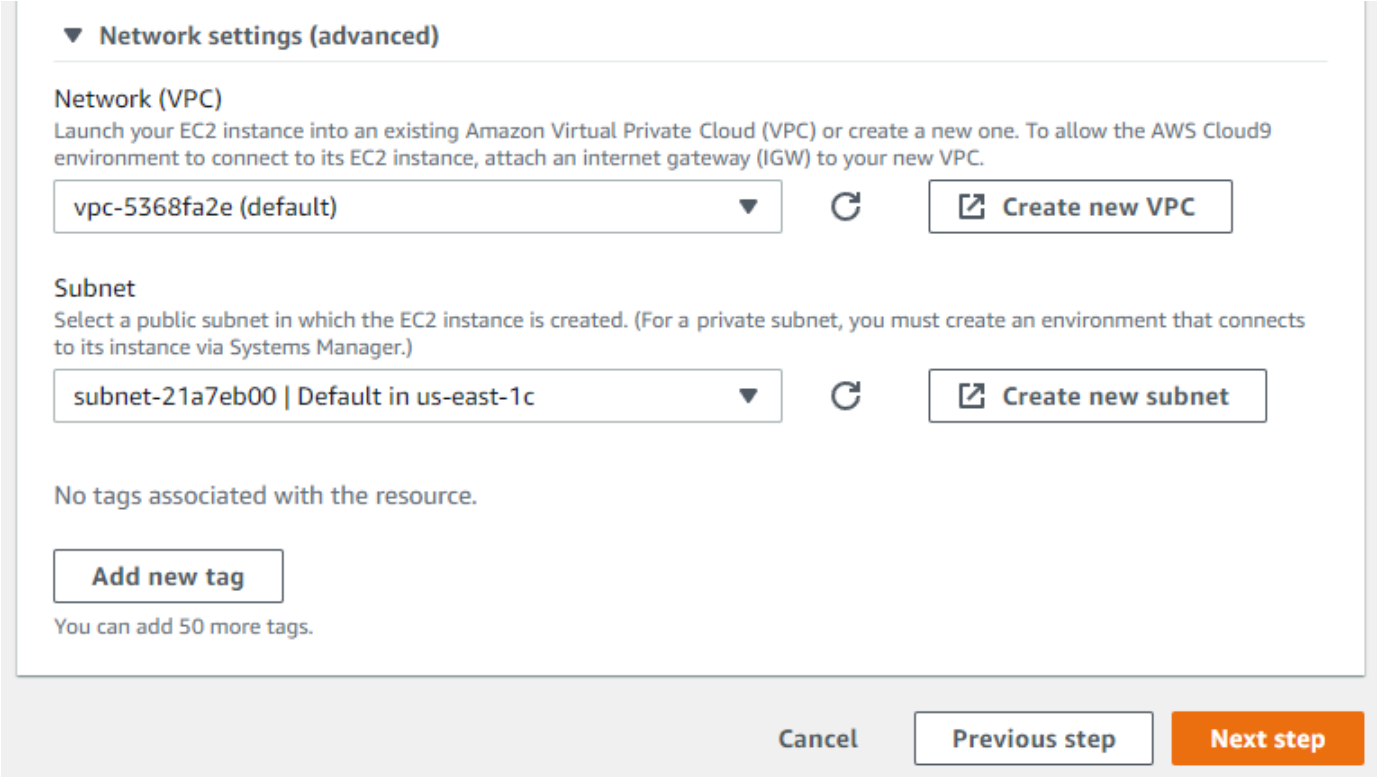
   Dans la section Plateforme, sélectionnez Amazon Linux 2 (recommandé).



6. Développez Paramètres réseau (avancés).

   Choisissez le VPC et l'un des sous-réseaux que vous avez utilisés lors de la création de votre cluster élastique.

   Choisissez Next step (Étape suivante).



7. Passez en revue votre AWS Cloud9 configuration.

   Si votre configuration est correcte, choisissez Create environment.

Étape 3 : Installation du shell Mongo

Une fois que votre AWS Cloud9 environnement est prêt, vous êtes prêt à vous connecter à votre cluster. Ensuite, installez le shell mongo dans votre AWS Cloud9 environnement que vous avez créé à l'étape 3. Le shell mongo est un utilitaire de ligne de commande que vous utilisez pour connecter et interroger votre cluster élastique.

Si votre AWS Cloud9 environnement est toujours ouvert après l'étape 3, revenez à cet environnement et passez à l'instruction 3. Si vous avez quitté votre AWS Cloud9 environnement, dans la AWS Cloud9 console, sous Vos environnements, recherchez l'environnement étiqueté avec le nom que vous avez défini à l'étape précédente. Choisissez Open IDE.

1. À l'invite de commande, créez le fichier de référentiel à l'aide de la commande suivante :

Example

```
echo -e "[mongodb-org-4.0] \nname=MongoDB Repository\nbaseurl=https://
repo.mongodb.org/yum/amazon/2013.03/mongodb-org/4.0/x86_64/\ngpgcheck=1 \nenabled=1
 \ngpgkey=https://www.mongodb.org/static/pgp/server-4.0.asc" | sudo tee /etc/
yum.repos.d/mongodb-org-4.0.repo
```

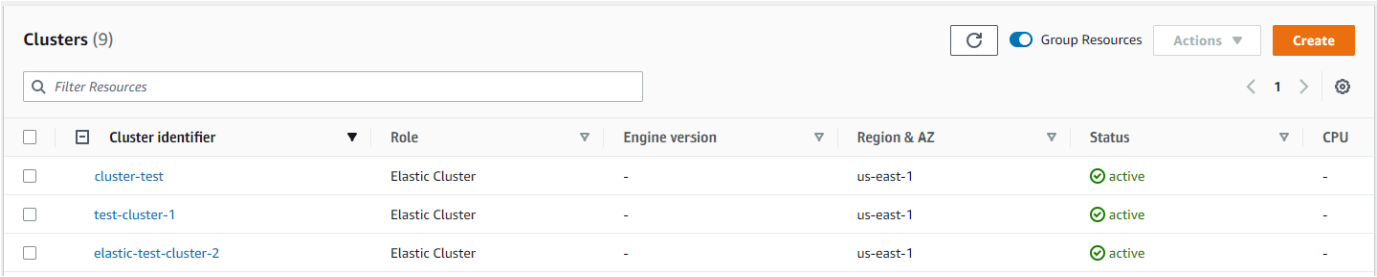
2. Une fois l'opération terminée, installez le shell mongo avec la commande suivante :

```
sudo yum install -y mongodb-org-shell
```

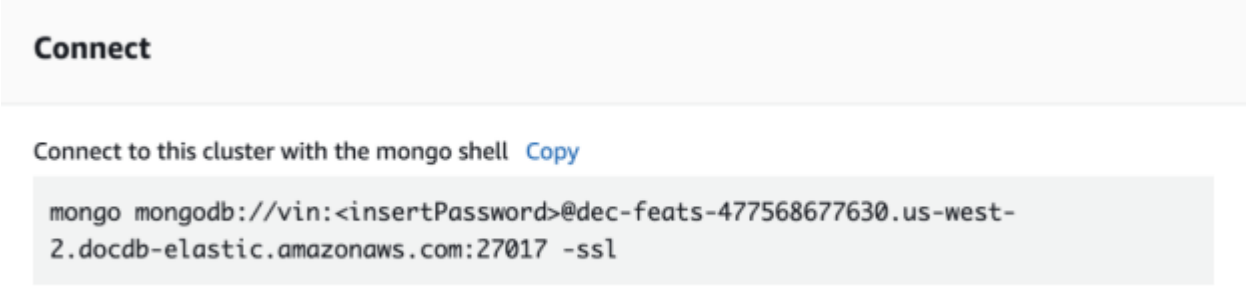
Étape 4 : Connectez-vous à votre nouveau cluster élastique

Connectez-vous à votre cluster à l'aide du shell mongo que vous avez installé à l'étape 4.

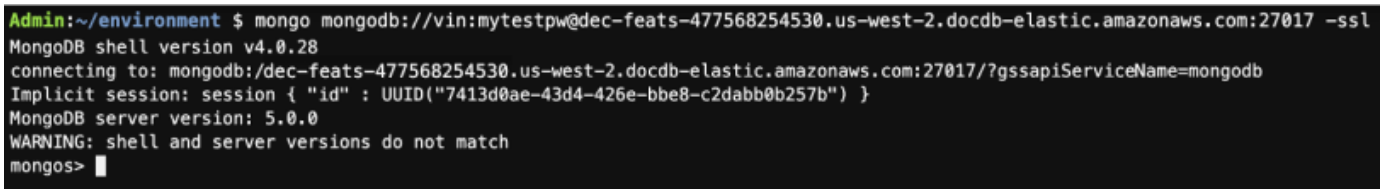
1. Sur la console de gestion Amazon DocumentDB, sous Clusters, localisez votre cluster. Triez par rôle pour afficher tous les clusters dotés du rôle Elastic Cluster.



2. Choisissez le cluster que vous avez créé en sélectionnant l'identifiant du cluster. Dans Connectivité et sécurité, copiez votre terminal et collez-le dans votre AWS Cloud9 environnement.



3. Une fois connecté, vous devriez voir le résultat suivant :



Étape 5 : Partagez votre collection ; insérez et interrogez des données

Les clusters élastiques ajoutent la prise en charge du sharding dans Amazon DocumentDB. Maintenant que vous êtes connecté à votre cluster, vous pouvez partager le cluster, insérer des données et exécuter quelques requêtes.

1. Pour partager une collection, entrez les informations suivantes :

   ```
   sh.shardCollection("db.Employee1" , { "Employeeid" : "hashed" })
   ```

2. Pour insérer un seul document, entrez les informations suivantes :

   ```
   db.Employee1.insert({"Employeeid":1, "Name":"Joe", "LastName": "Bruin",
   "level": 1 })
   ```

   La sortie suivante s'affiche :

   ```
   WriteResult({ "nInserted" : 1 })
   ```

3. Pour lire le document que vous avez écrit, entrez la `findOne()` commande (il renvoie un seul document) :

   ```
   db.Employee1.findOne()
   ```

   La sortie suivante s'affiche :

   Example

   ```
   {
   "_id" : ObjectId("61f344e0594fe1a1685a8151"),
   "EmployeeID" : 1,
   "Name" : "Joe",
   "LastName" : "Bruin",
   "level" : 1
   }
   ```

4. Pour effectuer quelques requêtes supplémentaires, considérez un cas d'utilisation d'un profil de jeu. Tout d'abord, insérez quelques entrées dans une collection intitulée « Employé ». Saisissez :

   Example

   ```
   db.Employee1.insertMany([
       { "Employeeid" : 1, "name" : "Matt", "lastname": "Winkle", "level": 12},
       { "Employeeid" : 2, "name" : "Frank", "lastname": "Chen", "level": 2},
       { "Employeeid" : 3, "name" : "Karen", "lastname": "William", "level": 7},
       { "Employeeid" : 4, "name" : "Katie", "lastname": "Schaper", "level": 3}
   ])
   ```

   La sortie suivante s'affiche :

   ```
   { "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4 ] }
   ```

5. Pour renvoyer tous les documents de la collection de profils, entrez la commande `find` () :

   ```
   db.Employee1.find()
   ```

   Les données que vous avez saisies à l'étape 4 s'affichent.

6. Pour interroger un seul document, incluez un filtre (par exemple : « Katie »). Saisissez :

   ```
   db.Employee1.find({name: "Katie"})
   ```

   La sortie suivante s'affiche :

   ```
   { "_id" : 4, "name" : "Katie", "lastname": "Schaper", "level": 3}
   ```

7. Pour rechercher un profil et le modifier, entrez la `findAndModify` commande. Dans cet exemple, le niveau « 14 » est attribué à l'employé « Matt » :

   Example

   ```
   db.Employee1.findAndModify({
       query: { "Employeeid" : 1, "name" : "Matt"},
       update: { "Employeeid" : 1, "name" : "Matt", "lastname" : "Winkle", "level" :
    14 }
   })
   ```

   Le résultat suivant s'affiche (notez que le niveau n'a pas encore changé) :

Example

```
{
    "_id" : 1,
    "name" : "Matt",
    "lastname" : "Winkle",
    "level" : 12,
}
```

8. Pour vérifier l'augmentation du niveau, entrez la requête suivante :

   ```
   db.Employee1.find({name: "Matt"})
   ```

   La sortie suivante s'affiche :

   ```
   { "_id" : 1, "name" : "Matt", "lastname" : "winkle", "level" : 14 }
   ```

Bonnes pratiques

Découvrez les bonnes pratiques d'utilisation de Amazon DocumentDB. Toutes les bonnes pratiques relatives aux clusters Amazon DocumentDB basés sur des instances s'appliquent également aux clusters élastiques. Cette section est mise à jour en continu à mesure que de nouvelles bonnes pratiques sont identifiées.

Rubriques

- Choix des clés de partition
- Gestion des connexions
- Collections non partagées
- Mise à l'échelle des clusters élastiques
- Surveillance des clusters élastiques

Choix des clés de partition

La liste suivante décrit les instructions relatives à la création de clés de partition.

- Utilisez une clé de hachage distribuée de manière uniforme pour répartir vos données sur toutes les partitions de votre cluster (évitez les touches de raccourci).

- Utilisez votre clé Shard dans toutes les demandes de lecture/mise à jour/suppression afin d'éviter les requêtes Scatter Gather.
- Évitez les clés de partition imbriquées lorsque vous effectuez des opérations de lecture/mise à jour/suppression.
- Lorsque vous effectuez des opérations par lots, définissez`ordered` cette valeur sur false afin que toutes les partitions puissent s'exécuter en parallel et améliorer les latences.

Gestion des connexions

La liste suivante décrit les instructions relatives à la gestion de vos connexions à votre base de données.

- Surveillez le nombre de vos connexions et la fréquence à laquelle de nouvelles connexions sont ouvertes et fermées.
- Répartissez vos connexions sur tous les sous-réseaux de la configuration de votre application. Si votre cluster est configuré sur plusieurs sous-réseaux mais que vous n'utilisez qu'un sous-ensemble de sous-réseaux, il se peut que vos connexions maximales soient limitées.

Collections non partagées

Ce qui suit décrit les directives relatives aux collections non partagées.

- Lorsque vous travaillez avec des collections non partitionnées, pour répartir la charge, essayez de conserver les collections non partitionnées les plus utilisées sur différentes bases de données. Les clusters élastiques Amazon DocumentDB placent les bases de données sur différentes partitions et colocalisent les collections non fragmentées de la même base de données sur la même partition.

Mise à l'échelle des clusters élastiques

La liste suivante décrit les instructions relatives à la mise à l'échelle de vos clusters élastiques.

- Les opérations de dimensionnement peuvent provoquer une brève période d'erreurs intermittentes de base de données et de réseau. Dans la mesure du possible, évitez le détartrage pendant les heures de pointe. Essayez d'utiliser la mise à l'échelle pendant les fenêtres de maintenance.
- Il est préférable d'augmenter ou de diminuer la capacité des partitions (modification du nombre de processeurs virtuels par partition) pour augmenter la capacité de calcul plutôt que d'augmenter

ou de diminuer le nombre de partitions, car cette méthode est plus rapide et entraîne des erreurs intermittentes de base de données et de réseau plus courtes.

- Lorsque vous anticipez une croissance, privilégiez l'augmentation du nombre de partitions plutôt que l'augmentation de la capacité des partitions. Cela vous permet de faire évoluer votre cluster en augmentant la capacité de la partition pour les scénarios nécessitant une mise à l'échelle rapide.
- Surveillez vos politiques relatives aux nouvelles tentatives côté client et réessayez avec un ralentissement et une instabilité exponentiels afin d'éviter de surcharger votre base de données en cas d'erreurs lors du dimensionnement.

Surveillance des clusters élastiques

La liste suivante décrit les directives relatives à la surveillance de vos clusters élastiques.

- Suivez lepeak-to-average ratio de vos indicateurs par fragment pour déterminer si vous générez un trafic irrégulier (utilisez une touche de raccourci ou une zone réactive). Les indicateurs clés permettant de suivrepeak-to-average les ratios sont les suivants :
  - `PrimaryInstanceCPUUtilization`
    - Cela peut être surveillé au niveau de chaque fragment.
    - Au niveau du cluster, vous pouvez surveiller l'inclinaison moyenne jusqu'à p99.
  - `PrimaryInstanceFreeableMemory`
    - Cela peut être surveillé au niveau de chaque fragment.
    - Au niveau du cluster, vous pouvez surveiller l'inclinaison moyenne jusqu'à p99.
  - `DatabaseCursorsMax`
    - Cela doit être surveillé au niveau de chaque fragment afin de déterminer l'inclinaison.
  - `Documents-Inserted/Updated/Returned/Deleted`
    - Cela doit être surveillé au niveau de chaque fragment afin de déterminer l'inclinaison.

Gestion des clusters élastiques

Pour gérer un cluster élastique Amazon DocumentDB, vous devez disposer d'une politique IAM avec les autorisations appropriées du plan de contrôle Amazon DocumentDB. Ces autorisations vous permettent de créer, de modifier et de supprimer des clusters. La FullAccess politique Amazon DocumentDB fournit toutes les autorisations requises pour administrer un cluster élastique Amazon DocumentDB.

Les rubriques suivantes montrent comment effectuer différentes tâches lorsque vous travaillez avec des clusters élastiques Amazon DocumentDB.

Rubriques

Modification des configurations de clusters élastiques

Dans cette section, nous expliquons comment modifier Elastic Cluster, en utilisant AWS Management Console ou AWS CLI en suivant les instructions suivantes.

L'une des principales utilisations de la modification du cluster consiste à redimensionner les partitions en augmentant ou en diminuant le nombre de partitions et/ou la capacité de calcul des partitions.

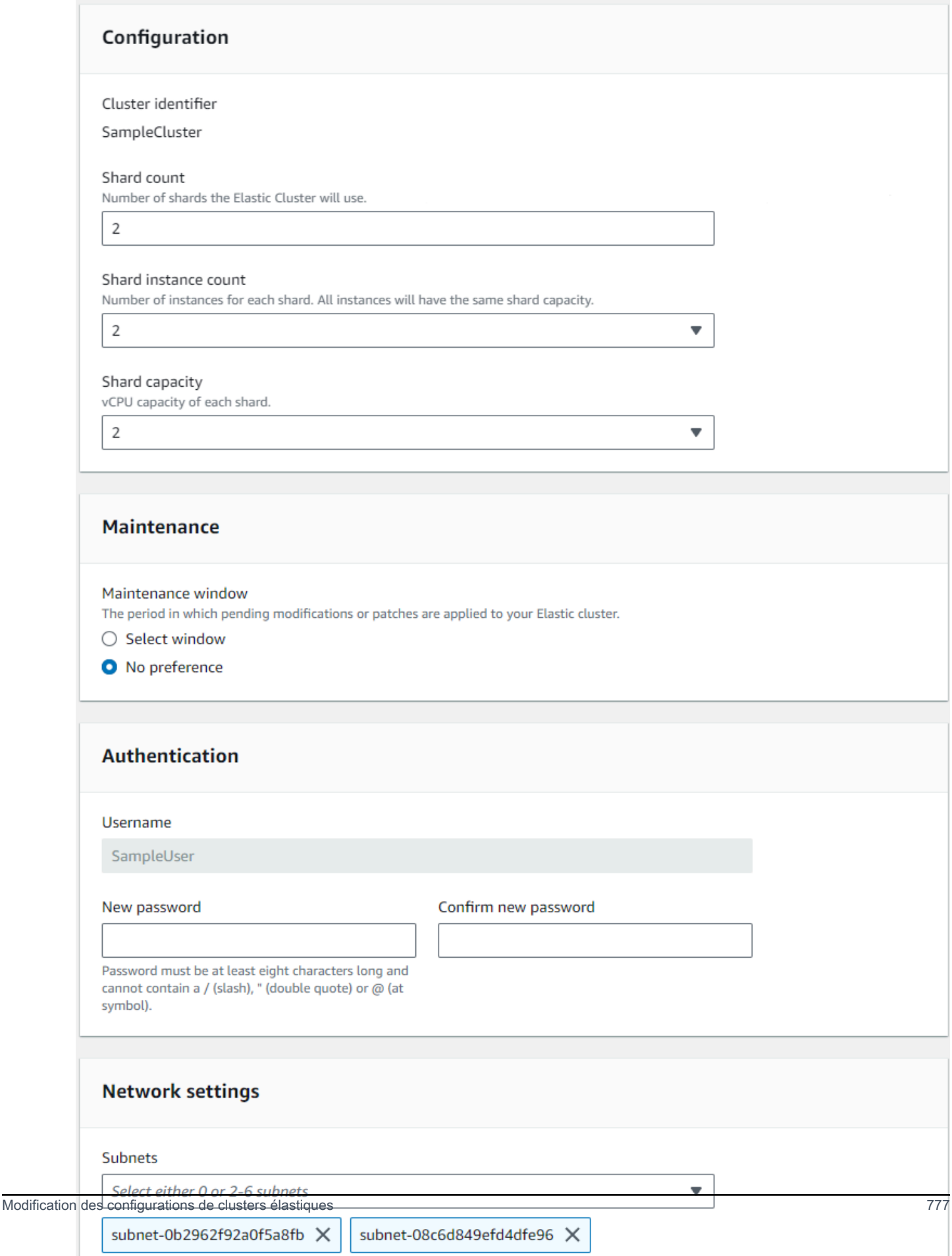
Using the AWS Management Console

Pour modifier la configuration d'un cluster élastique à l'aide de AWS Management Console :

1. Connectez-vous à la console Amazon DocumentDB AWS Management Consoleet ouvrez-la.
2. Dans le panneau de navigation, choisissez Clusters.

   > **Tip**
   >
   > Si le volet de navigation n'apparaît pas sur le côté gauche de votre écran, choisissez l'icône de menu dans le coin supérieur gauche du volet de navigation.

3. Choisissez le nom du cluster que vous souhaitez modifier dans la colonne Identifiant du cluster.
4. Sélectionnez Modifier.
5. Modifiez les champs que vous souhaitez modifier, puis sélectionnez Modifier le cluster.


Configuration
Cluster identifier
SampleCluster
Shard count
Number of shards the Elastic Cluster will use.
2
Shard instance count
Number of instances for each shard. All instances will have the same shard capacity.
2
Shard capacity
vCPU capacity of each shard.
2
Maintenance
Maintenance window
The period in which pending modifications or patches are applied to your Elastic cluster.
Select window
No preference
Authentication
Username
SampleUser
New password
Confirm new password
Password must be at least eight characters long and cannot contain a / (slash), " (double quote) or @ (at symbol).
Network settings
Subnets
Select either 0 or 2-6 subnets
subnet-0b2962f92a0f5a8fb
subnet-08c6d849efd4dfe96
VPC security groups

> **Note**
>
> Vous pouvez également accéder à la boîte de dialogue Modifier le cluster en accédant à la page Clusters, en cochant la case à côté de votre cluster, en choisissant Actions, puis Modifier.

Using the AWS CLI

Pour modifier une configuration de cluster élastique à l'aide de AWS CLI, utilisez l'`update-cluster`opération avec les paramètres suivants :

- **`--cluster-arn`** : obligatoire. Identifiant ARN du cluster que vous souhaitez modifier.
- **`--shard-capacity`**—Facultatif. Le nombre de vCPU assignés à chaque partition. Le maximum est de 64. Les valeurs autorisées sont 2, 4, 8, 16, 32, 64.
- **`--shard-count`**—Facultatif. Le nombre de partitions attribuées au cluster. Le maximum est de 32.
- **`--shard-instance`**-Nombre : facultatif. Le nombre d'instances de répliques s'appliquant à toutes les partitions de ce cluster. Le maximum est de 16.
- **`--auth-type`**—Facultatif. Type d'authentification utilisé pour déterminer où récupérer le mot de passe utilisé pour accéder au cluster élastique. Les types valides sont `PLAIN_TEXT` ou`SECRET_ARN`.
- **`--admin-user-password`**—Facultatif. Le mot de passe associé à l'utilisateur administrateur.
- **`--vpc-security-group-ids`**—Facultatif. Configurez une liste des groupes de sécurité Amazon EC2 et Amazon Virtual Private Cloud (VPC) à associer à ce cluster.
- **`--preferred-maintenance-window`**—Facultatif. Configurer la plage horaire hebdomadaire pendant laquelle la maintenance du système peut avoir lieu, en temps universel coordonné (UTC)

  Le format est :`ddd:hh24:mi-ddd:hh24:mi`. Jours valides (ddd) : lundi, mardi, mercredi, jeudi, vendredi, samedi, dimanche

  La valeur par défaut est une fenêtre de 30 minutes sélectionnée au hasard sur une période de 8 heures pour chaque région Amazon Web Services, survenant un jour aléatoire de la semaine.

  Fenêtre minimale de 30 minutes.
- **`--subnet-ids`**—Facultatif. Configurez les identifiants de sous-réseau.

Dans les exemples suivants, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations.

Pour Linux, macOS ou Unix :

```
aws docdb-elastic update-cluster \
    --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 \
    --shard-capacity 8 \
    --shard-count 4 \
    --shard-instance-count 3 \
    --admin-user-password testPassword \
    --vpc-security-group-ids ec-65f40350 \
    --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

Pour Windows :

```
aws docdb-elastic update-cluster ^
    --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 ^
    --shard-capacity 8 ^
    --shard-count 4 ^
    --shard-instance-count 3 ^
    --admin-user-password testPassword ^
    --vpc-security-group-ids ec-65f40350 ^
    --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

Pour surveiller l'état du cluster élastique après votre modification, consultez la section Surveillance d'un cluster élastique.

Surveillance d'un cluster élastique

Dans cette section, nous expliquons comment surveiller votre cluster élastique, en utilisant AWS Management Console ou AWS CLI en suivant les instructions suivantes.

Using the AWS Management Console

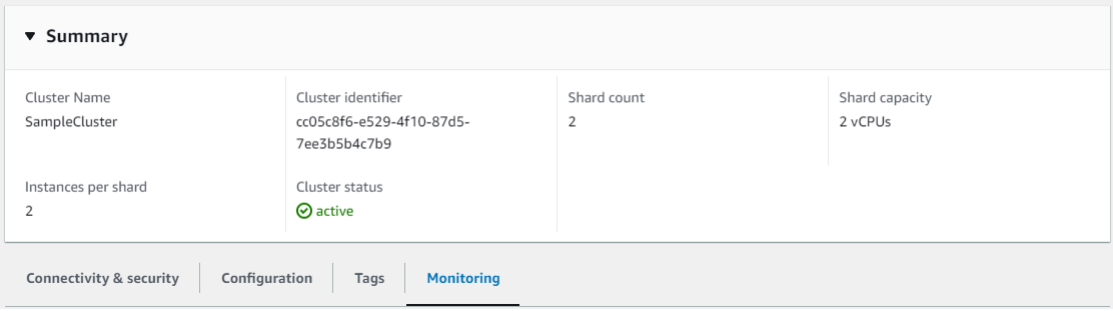
Pour surveiller une configuration de cluster élastique à l'aide de AWS Management Console :

1. Connectez-vous à la console Amazon DocumentDB AWS Management Consoleet ouvrez-la.

2. Dans le panneau de navigation, choisissez Clusters.

> **Tip**
>
> Si le volet de navigation n'apparaît pas sur le côté gauche de votre écran, choisissez l'icône de menu dans le coin supérieur gauche du volet de navigation.

3. Choisissez le nom du cluster que vous souhaitez surveiller dans la colonne Identifiant du cluster.
4. Sélectionnez l'onglet Monitoring (Surveillance).



Un certain nombre de graphiques d'Amazon CloudWatch sont affichés pour les catégories de surveillance suivantes :

- Utilisation des ressources
- Débit
- Latence
- Opérations
- Système

Vous pouvez également accéder à Amazon CloudWatch via le AWS Management Console pour configurer votre propre environnement de surveillance pour vos clusters élastiques.

Using the AWS CLI

Pour surveiller une configuration de cluster élastique spécifique à l'aide de AWS CLI, utilisez l'`get-cluster`opération avec les paramètres suivants :

- **`--cluster-arn`** : obligatoire. Identifiant ARN du cluster pour lequel vous souhaitez obtenir des informations.

Dans les exemples suivants, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations.

Pour Linux, macOS ou Unix :

```
aws docdb-elastic get-cluster \
    --cluster-arn arn:aws:docdb-elastic:us-west-2:123456789012:cluster:/68ffcdf8-e3af-40a3-91e4-24736f2dacc9
```

Pour Windows :

```
aws docdb-elastic get-cluster ^
    --cluster-arn arn:aws:docdb:-elastic:us-west-2:123456789012:cluster:/68ffcdf8-e3af-40a3-91e4-24736f2dacc9
```

Le résultat de cette opération ressemble à ce qui suit :

```
"cluster": {
        ...
        "clusterArn": "arn:aws:docdb-elastic:us-
west-2:123456789012:cluster:/68ffcdf8-e3af-40a3-91e4-24736f2dacc9",
        "clusterEndpoint": "stretch-11-477568257630.us-east-1.docdb-
elastic.amazonaws.com",
        "readerEndpoint": "stretch-11-477568257630-ro.us-east-1.docdb-
elastic.amazonaws.com",
        "clusterName": "stretch-11",
        "shardCapacity": 2,
        "shardCount": 3,
        "shardInstanceCount: 5,
        "status": "ACTIVE",
        ...
 }
```

Pour plus d'informations, consultez le document `DescribeClusterSnapshot` de référence de l'API de gestion des ressources Amazon DocumentDB.

Pour afficher les détails de tous les clusters élastiques à l'aide de AWS CLI, utilisez l'`list-clusters`opération avec les paramètres suivants :

- **`--next-token`**—Facultatif. Si le nombre d'éléments en sortie (`--max-results`) est inférieur au nombre total d'éléments renvoyés par les appels d'API sous-jacents, la sortie inclut un

code `NextToken` que vous pouvez transmettre dans une commande suivante pour extraire le prochain ensemble d'éléments.

- **`--max-results`**—Facultatif. Le nombre total d'éléments à renvoyer dans la sortie de la commande. S'il existe plus de résultats que la `max-results` valeur spécifiée, un jeton de pagination (`next-token`) est inclus dans la réponse afin que les résultats restants puissent être récupérés.
  - Par défaut : 100
  - Minimum 20, maximum 100

Dans les exemples suivants, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations.

Pour Linux, macOS ou Unix :

```
aws docdb-elastic list-clusters \
    --next-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ== \
    --max-results 2
```

Pour Windows :

```
aws docdb-elastic list-clusters ^
    --next-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ== ^
    --max-results 2
```

Le résultat de cette opération ressemble à ce qui suit :

```
{
   "Clusters": [
      {
         "ClusterIdentifier":"mycluster-1",
         "ClusterArn":"arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster"
         "Status":"available",
         "ClusterEndpoint":"sample-cluster.sharded-cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com"
       }
       {
         "ClusterIdentifier":"mycluster-2",
         "ClusterArn":"arn:aws:docdb:us-west-2:987654321098:sharded-cluster:sample-
cluster"
```

```
         "Status":"available",
         "ClusterEndpoint":"sample-cluster2.sharded-cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com"
       }
   ]
}
```

Supprimer un cluster élastique

Dans cette section, nous expliquons comment supprimer un cluster élastique, en utilisant AWS Management Console ou AWS CLI en suivant les instructions suivantes.

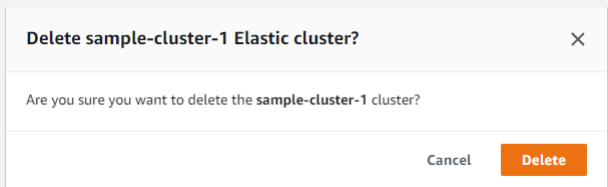
Using the AWS Management Console

Pour supprimer une configuration de cluster élastique à l'aide de AWS Management Console :

1. Connectez-vous à la console Amazon DocumentDB AWS Management Consoleet ouvrez-la.
2. Dans le panneau de navigation, choisissez Clusters.

> **Tip**
>
> Si le volet de navigation n'apparaît pas sur le côté gauche de votre écran, choisissez l'icône de menu dans le coin supérieur gauche du volet de navigation.

3. Dans le tableau de liste des clusters, cochez la case située à gauche du nom du cluster que vous souhaitez supprimer, puis choisissez Actions. Dans le menu déroulant, choisissez Delete (Supprimer).
4. Dans le cluster élastique Delete « cluster-name » ? dans une boîte de dialogue, choisissez Supprimer.



La suppression du cluster prend plusieurs minutes. Pour surveiller l'état du cluster, consultez la section Surveillance de l'état d'un cluster Amazon DocumentDB.

Using the AWS CLI

Pour supprimer un cluster élastique à l'aide de AWS CLI, utilisez l'`delete-cluster`opération avec les paramètres suivants :

- **`--cluster-arn`** : obligatoire. Identifiant ARN du cluster que vous souhaitez supprimer.
- **`--no-skip-final-backup`**—Facultatif. Si vous souhaitez effectuer une sauvegarde finale, vous devez inclure ce paramètre dans le nom de la sauvegarde finale. Vous devez inclure soit `--final-backup-identifier` ou `--skip-final-backup`.
- **`--skip-final-backup`**—Facultatif. Utilisez ce paramètre uniquement si vous ne souhaitez pas effectuer de sauvegarde finale avant de supprimer votre cluster. Le paramètre par défaut consiste à prendre un instantané final.

Les exemples de AWS CLI code suivants suppriment un cluster dont l'ARN est arn:aws:docdb:us-west- 2:123456789012:sharded-cluster:sample-cluster avec une sauvegarde finale.

Dans l'exemple suivant, remplacez chaque *espace réservé saisi par l'utilisateur* par vos propres informations.

Pour Linux, macOS ou Unix :

```
aws docdb-elastic delete-cluster \
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster \
    --no-skip-final-backup \
    --final-backup-identifier finalArnBU-arn:aws:docdb:us-
west-2:123456789012:sharded-cluster:sample-cluster
```

Pour Windows :

```
aws docdb-elastic delete-cluster ^
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster ^
    --no-skip-final-backup ^
    --final-backup-identifier finalArnBU-arn:aws:docdb:us-
west-2:123456789012:sharded-cluster:sample-cluster
```

Les exemples de AWS CLI code suivants suppriment un cluster dont l'ARN est arn:aws:docdb:us-west- 2:123456789012:sharded-cluster:sample-cluster sans effectuer de sauvegarde finale.

Dans les exemples suivants, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations.

Pour Linux, macOS ou Unix :

```
aws docdb-elastic delete-cluster \
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster \
    --skip-final-backup \
```

Pour Windows :

```
aws docdb-elastic delete-cluster ^
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster ^
    --skip-final-backup ^
```

Le résultat de l'`delete-cluster`opération est un affichage du cluster que vous supprimez.

La suppression du cluster prend plusieurs minutes. Pour surveiller l'état du cluster, consultez la section Surveillance de l'état d'un cluster Amazon DocumentDB.

Gestion des instantanés de clusters élastiques

Des instantanés manuels peuvent être pris après la création d'un cluster élastique. Les sauvegardes automatisées sont créées au moment où le snapshot du cluster élastique est créé.

> **Note**
>
> Votre cluster élastique doit être dans l'`Available`état requis pour qu'un instantané soit pris manuellement.

Cette section explique comment créer, afficher, restaurer et supprimer des instantanés d'Elastic Cluster.

Les rubriques suivantes expliquent comment effectuer différentes tâches lorsque vous travaillez avec des instantanés de clusters élastiques Amazon DocumentDB.

Rubriques

Création d'un instantané manuel d'un cluster élastique

Dans cette section, nous expliquons comment créer un instantané manuel d'un cluster élastique, en utilisant AWS Management Console ou AWS CLI en suivant les instructions suivantes.

Using the AWS Management Console

Pour créer un instantané manuel d'un cluster élastique à l'aide de AWS Management Console :

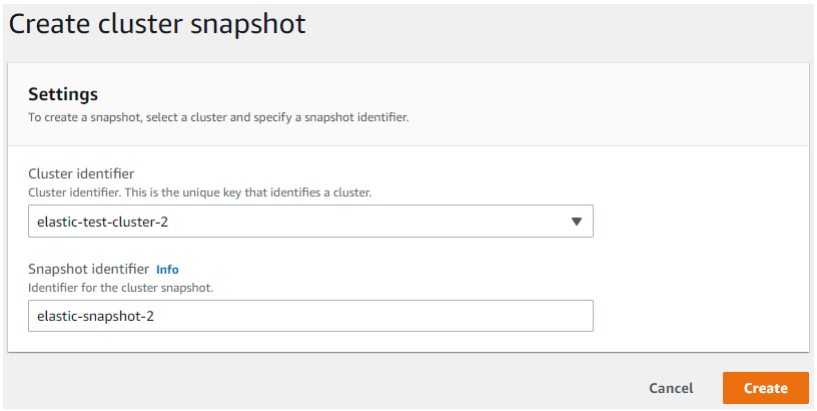
1. Connectez-vous à la console Amazon DocumentDB AWS Management Consoleet ouvrez-la.
2. Dans le panneau de navigation, choisissez Snapshots (Instantanés).

   > **Tip**
   >
   > Si le volet de navigation n'apparaît pas sur le côté gauche de votre écran, choisissez l'icône de menu dans le coin supérieur gauche du volet de navigation.

3. Sur la page Instantanés, choisissez Créer.
4. Sur la page Créer un instantané de cluster, dans le champ Identifiant du cluster, choisissez votre cluster élastique dans la liste déroulante.

   Dans le champ Snapshot identifier, entrez un identifiant unique pour votre cluster élastique.

   Choisissez Créer.



> **Note**
>
> Vous pouvez également accéder à la boîte de dialogue Créer un instantané de cluster en accédant à la page Clusters, en cochant la case à côté de votre cluster, puis en choisissant Actions, puis Prendre un instantané.

Le snapshot de votre cluster Elastic est en cours de provisionnement. Cette opération peut prendre jusqu'à quelques minutes. Vous pouvez afficher et restaurer à partir de votre instantané lorsque l'état s'affiche, comme `Available` dans la liste des instantanés.

Using the AWS CLI

Pour créer un instantané manuel d'un cluster élastique à l'aide de AWS CLI, utilisez l'`create-cluster-snapshot`opération avec les paramètres suivants :

- **`--snapshot-name`** : obligatoire. Nom de l'instantané du cluster que vous souhaitez créer.
- **`--cluster-arn`** : obligatoire. Identifiant ARN du cluster dont vous souhaitez créer un instantané.

Dans les exemples suivants, remplacez chaque *`espace réservé pour l'entrée utilisateur`* par vos propres informations.

Pour Linux, macOS ou Unix :

```
aws docdb-elastic create-cluster-snapshot \
    --snapshot-name sample-snapshot-1 \
```

```
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster
```

Pour Windows :

```
aws docdb-elastic create-cluster-snapshot ^
    --snapshot-name sample-snapshot-1 ^
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster
```

Affichage d'un instantané d'un cluster élastique

Dans cette section, nous expliquons comment afficher les informations des instantanés d'Elastic Cluster, à l'aide des instructions suivantes AWS Management Console ou AWS CLI en suivant les instructions suivantes.

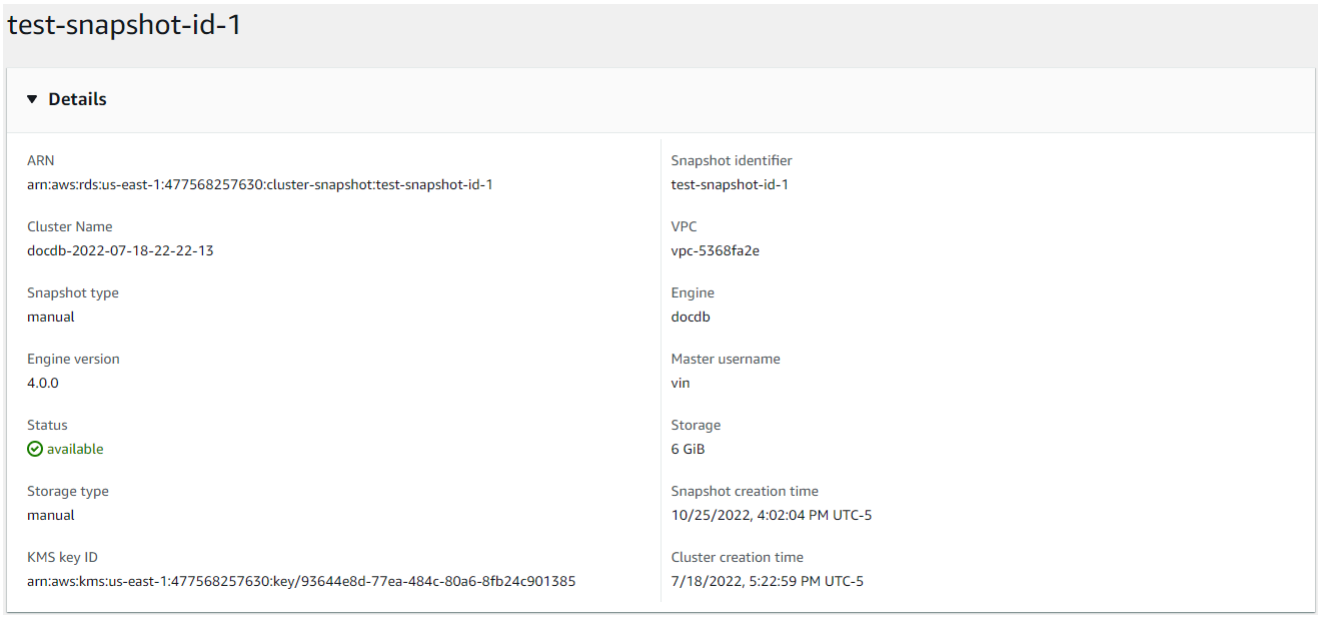
Using the AWS Management Console

Pour afficher les informations relatives à un instantané d'un cluster élastique spécifique à l'aide de AWS Management Console :

1. Connectez-vous à la console Amazon DocumentDB AWS Management Consoleet ouvrez-la.
2. Dans le panneau de navigation, choisissez Snapshots (Instantanés).

   > **Tip**
   >
   > Si le volet de navigation n'apparaît pas sur le côté gauche de votre écran, choisissez l'icône de menu dans le coin supérieur gauche du volet de navigation.

3. Sur la page Instantanés, choisissez votre instantané dans la liste en cliquant sur le nom dans la colonne Identifiant du cliché.
4. Consultez les informations de votre instantané dans Détails.



Using the AWS CLI

Pour afficher les informations relatives à un instantané de cluster élastique spécifique à l'aide de AWS CLI, utilisez l'`get-cluster-snapshot`opération avec les paramètres suivants :

- **`--snapshot-arn`** : obligatoire. Identifiant ARN de l'instantané pour lequel vous souhaitez obtenir des informations.

Dans les exemples suivants, remplacez chaque *`espace réservé pour l'entrée utilisateur`* par vos propres informations.

Pour Linux, macOS ou Unix :

```
aws docdb-elastic get-cluster-snapshot \
    --snapshot-arn sampleResourceName
```

Pour Windows :

```
aws docdb-elastic get-cluster-snapshot ^
    --snapshot-arn sampleResourceName
```

Pour afficher les informations relatives à un instantané de cluster élastique spécifique à l'aide de AWS CLI, utilisez l'`get-cluster-snapshot`opération avec les paramètres suivants :

- **`--snapshot-arn`** : obligatoire. Identifiant ARN de l'instantané pour lequel vous souhaitez obtenir des informations.

Dans les exemples suivants, remplacez chaque *`espace réservé pour l'entrée utilisateur`* par vos propres informations.

Pour Linux, macOS ou Unix :

```
aws docdb-elastic get-cluster-snapshot \
    --snapshot-arn sampleResourceName
```

Pour Windows :

```
aws docdb-elastic get-cluster-snapshot ^
    --snapshot-arn sampleResourceName
```

Pour afficher des informations sur tous les instantanés d'Elastic Cluster à l'aide de AWS CLI, utilisez l'`list-cluster-snapshots`opération avec les paramètres suivants :

- **`--snapshot-type`**—Facultatif. Type de snapshots de cluster à renvoyer. Vous pouvez spécifier l'une des valeurs suivantes :
  - `automated`- Renvoie tous les instantanés de cluster qu'Amazon DocumentDB a automatiquement créés pour AWS votre compte.
  - `manual`- Renvoie tous les instantanés de cluster que vous avez créés manuellement pour votre AWS compte.
  - `shared`- Renvoie tous les instantanés de cluster manuels qui ont été partagés sur votre AWS compte.
  - `public`- Renvoie tous les instantanés du cluster marqués comme publics.
- **`--next-token`**—Facultatif. Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement les enregistrements au-delà de ce jeton, jusqu'à la valeur spécifiée par`max-results`.
- **`--max-results`**—Facultatif. Le nombre maximum de résultats à inclure dans la réponse. S'il existe plus de résultats que la `max-results` valeur spécifiée, un jeton de pagination (`next-token`) est inclus dans la réponse afin que les résultats restants puissent être récupérés.

- Par défaut : 100
- Minimum 20, maximum 100

Dans les exemples suivants, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations.

Pour Linux, macOS ou Unix :

```
aws docdb-elastic list-cluster-snapshots \
    --snapshot-type  value \
    --next-token  value \
    --max-results 50
```

Pour Windows :

```
aws docdb-elastic list-cluster-snapshots ^
    --snapshot-type  value ^
    --next-token  value ^
    --max-results 50
```

Restauration d'un cluster élastique à partir d'un instantané

Dans cette section, nous expliquons comment restaurer un cluster élastique à partir d'un instantané, en utilisant AWS Management Console ou AWS CLI en suivant les instructions suivantes.

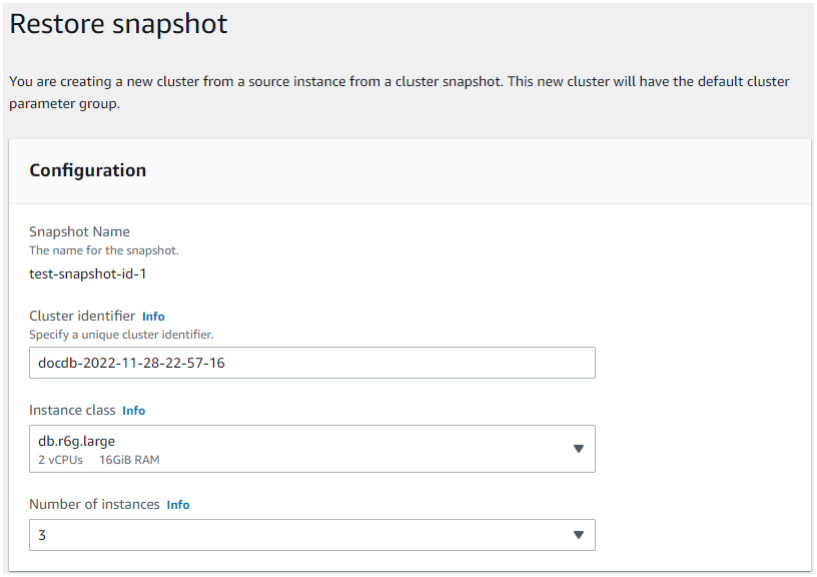
Using the AWS Management Console

Pour restaurer un cluster élastique à partir d'un instantané à l'aide de AWS Management Console :

1. Connectez-vous à la console Amazon DocumentDB AWS Management Consoleet ouvrez-la.
2. Dans le panneau de navigation, choisissez Snapshots (Instantanés).

> **Tip**
>
> Si le volet de navigation n'apparaît pas sur le côté gauche de votre écran, choisissez l'icône de menu dans le coin supérieur gauche du volet de navigation.

3. Cliquez sur le bouton situé à gauche de l'instantané, que vous souhaitez utiliser pour restaurer un cluster, dans la colonne Identifiant du cliché.
4. Choisissez Actions, puis Restaurer.



5. Sur la page Restaurer un instantané, entrez le nom du nouveau cluster dans le champ Identifiant du cluster.

> **Note**
>
> Pour toute restauration manuelle de snapshots, vous devez créer un nouveau cluster.

6. Dans le champ Virtual Private Cloud (VPC), sélectionnez un VPC dans la liste déroulante.
7. Pour les sous-réseaux et les groupes de sécurité VPC, vous pouvez utiliser les valeurs par défaut ou sélectionner trois sous-réseaux de votre choix et jusqu'à trois groupes de sécurité VPC (un minimum).
8. Si la configuration du cluster vous convient, choisissez Restore cluster (Restaurer le cluster) et patientez pendant la restauration du cluster.

Using the AWS CLI

Pour restaurer un cluster élastique à partir d'un instantané à l'aide de AWS CLI, utilisez l'`restore-cluster-from-snapshot`opération avec les paramètres suivants :

- **`--cluster-name`** : obligatoire. Nom actuel du cluster élastique tel qu'il a été saisi lors de sa création ou modifié pour la dernière fois.

- **`--snapshot-arn`** : obligatoire. Identifiant ARN du snapshot utilisé pour restaurer le cluster.
- **`--vpc-security-group-ids`**—Facultatif. Un ou plusieurs groupes de sécurité Amazon EC2 et Amazon Virtual Private Cloud (VPC) à associer au cluster.
- **`--kms-key-id`**—Facultatif. Configurez l'identifiant de clé KMS pour un cluster chiffré.

  L'identifiant de clé KMS est l'Amazon Resource Name (ARN) de la clé de AWS KMS chiffrement. Si vous créez un cluster en utilisant le même compte Amazon Web Services qui possède la clé de chiffrement KMS utilisée pour chiffrer le nouveau cluster, vous pouvez utiliser l'alias de clé KMS au lieu de l'ARN pour la clé de chiffrement KMS.

  Si aucune clé de chiffrement n'est spécifiée dans KmsKeyId et si le `StorageEncrypted` paramètre est vrai, Amazon DocumentDB utilise votre clé de chiffrement par défaut.
- **`--subnet-ids`**—Facultatif. Identifiants de sous-réseaux réseau.

Dans l'exemple suivant, remplacez chaque *espace réservé saisi par l'utilisateur* par vos propres informations.

Pour Linux, macOS ou Unix :

```
aws docdb-elastic restore-cluster-from-snapshot \
     --cluster-name elastic-sample-cluster \
     --snapshot-arn sampleResourceName \
     --vpc-security-group-ids value ec-65f40350 \
     --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 \
     --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

Pour Windows :

```
aws docdb-elastic restore-cluster-from-snapshot ^
     --cluster-name elastic-sample-cluster ^
     --snapshot-arn sampleResourceName ^
     --vpc-security-group-ids value ec-65f40350 ^
     --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 ^
     --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

Copier un instantané d'un cluster élastique

Dans Amazon DocumentDB, vous pouvez copier des instantanés de clusters élastiques manuels et automatiques au sein d'une même région et d'un même compte. Dans cette section, nous expliquons comment copier un instantané d'un cluster élastique, en utilisant le AWS Management Console ou AWS CLI.

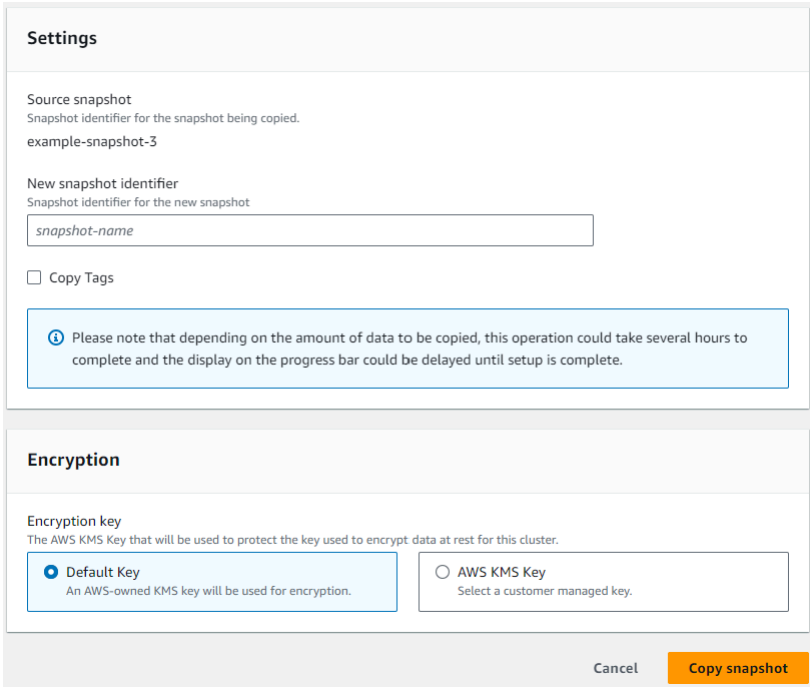
Using the AWS Management Console

Pour copier un instantané d'un cluster élastique à l'aide de AWS Management Console :

1. Connectez-vous à la console Amazon DocumentDB AWS Management Consoleet ouvrez-la.
2. Dans le panneau de navigation, choisissez Snapshots (Instantanés).

> **Tip**
>
> Si le volet de navigation n'apparaît pas sur le côté gauche de votre écran, choisissez l'icône de menu dans le coin supérieur gauche du volet de navigation.

3. Cliquez sur le bouton situé à gauche de l'instantané que vous souhaitez copier dans la colonne Identifiant du cliché.
4. Choisissez Actions, puis Copier.



5. Pour Nouvel identifiant de capture d'écran, entrez le nom du nouveau cliché.

6. Pour Copier les balises, cochez la case si vous souhaitez copier toutes les balises de l'instantané du cluster élastique source vers l'instantané du cluster élastique cible.
7. Pour le chiffrement, choisissez une clé AWS KMS par défaut ou une clé KMS de votre choix. La deuxième option vous permet de sélectionner une clé KMS existante que vous avez déjà créée ou d'en créer une nouvelle.
8. Choisissez Copier un instantané lorsque vous avez terminé.

Using the AWS CLI

Pour copier un instantané d'un cluster élastique à l'aide de AWS CLI, utilisez l'`copy-cluster-snapshot`opération avec les paramètres suivants :

- **`--source-db-cluster-snapshot-identifier`** : obligatoire. Identifiant de l'instantané du cluster élastique existant en cours de copie. L'instantané du cluster élastique doit exister et être dans l'état disponible. Si vous copiez l'instantané vers un autre Région AWS, cet identifiant doit être au format ARN de la source Région AWS. Ce paramètre n'est pas sensible à la casse.
- **`--target-db-cluster-snapshot-identifier`** : obligatoire. Identifiant du nouvel instantané de cluster élastique à créer à partir de l'instantané de cluster existant. Ce paramètre n'est pas sensible à la casse.

  Contraintes relatives au nom des instantanés cibles :

  - Ne peut pas être le nom d'un instantané existant.
  - La longueur est de [1 à 63] lettres, chiffres ou traits d'union.
  - Le premier caractère doit être une lettre.
  - Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

Dans les exemples suivants, remplacez chaque *`espace réservé pour l'entrée utilisateur`* par vos propres informations.

Pour Linux, macOS ou Unix :

```
aws docdb-elastic copy-cluster-snapshot \
     --source-cluster-snapshot-arn <sample ARN> \
     --target-cluster-snapshot-name my-target-copied-snapshot
```

Pour Windows :

```
aws docdb-elastic copy-cluster-snapshot ^
     --source-cluster-snapshot-arn <sample ARN> ^
     --target-cluster-snapshot-name my-target-copied-snapshot
```

Suppression d'un instantané d'un cluster élastique

Dans cette section, nous expliquons comment supprimer un instantané d'un cluster élastique, en utilisant le AWS Management Console ou AWS CLI.

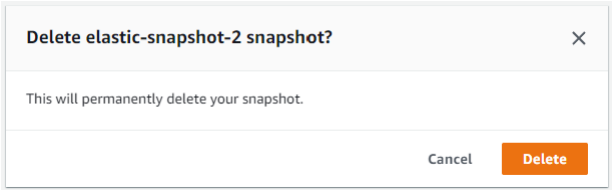
Using the AWS Management Console

Pour restaurer un cluster élastique à partir d'un instantané à l'aide de AWS Management Console :

1. Connectez-vous à la console Amazon DocumentDB AWS Management Consoleet ouvrez-la.
2. Dans le panneau de navigation, choisissez Snapshots (Instantanés).

> **Tip**
>
> Si le volet de navigation n'apparaît pas sur le côté gauche de votre écran, choisissez l'icône de menu dans le coin supérieur gauche du volet de navigation.

3. Cliquez sur le bouton situé à gauche de l'instantané, que vous souhaitez utiliser pour restaurer un cluster, dans la colonne Identifiant du cliché.
4. Choisissez Actions, puis Supprimer.



5. Dans la boîte de dialogue Supprimer l'instantané « nom du cliché », choisissez Supprimer.

Using the AWS CLI

Pour supprimer un instantané d'un cluster élastique à l'aide de AWS CLI, utilisez l'`delete-cluster-snapshot`opération avec les paramètres suivants :

- **`--snapshot-arn`** : obligatoire. Identifiant ARN du snapshot utilisé pour restaurer le cluster.

Dans l'exemple suivant, remplacez chaque *espace réservé saisi par l'utilisateur* par vos propres informations.

Pour Linux, macOS ou Unix :

```
aws docdb-elastic delete-cluster-snapshot \
     --snapshot-arn sampleResourceName
```

Pour Windows :

```
aws docdb-elastic delete-cluster-snapshot ^
     --snapshot-arn sampleResourceName
```

Gestion d'une sauvegarde automatique des instantanés d'un cluster Elastic

Amazon DocumentDB prend des instantanés quotidiens de vos clusters élastiques. Vous pouvez spécifier la fenêtre de sauvegarde préférée et la période de conservation des sauvegardes dans une configuration de snapshot Elastic Cluster nouvelle ou existante. Dans cette section, nous expliquons comment définir les paramètres de sauvegarde automatique dans un instantané de cluster élastique, en utilisant le AWS Management Console ou AWS CLI.

Using the AWS Management Console

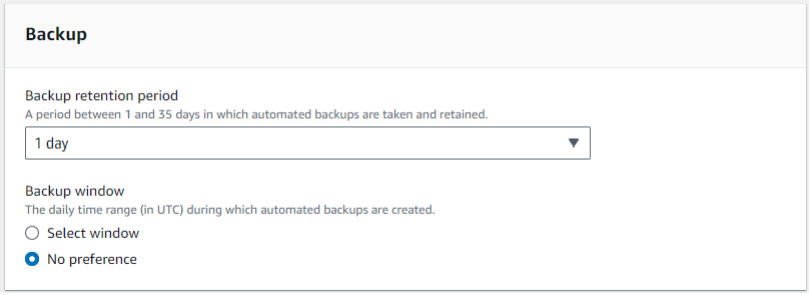
Pour configurer une sauvegarde automatique pour un nouvel instantané d'un cluster Elastic à l'aide de AWS Management Console :

1. Connectez-vous à la console Amazon DocumentDB AWS Management Consoleet ouvrez-la.
2. Dans le panneau de navigation, choisissez Clusters.

   > **Tip**
   >
   > Si le volet de navigation n'apparaît pas sur le côté gauche de votre écran, choisissez l'icône de menu dans le coin supérieur gauche du volet de navigation.

3. Cliquez sur le bouton situé à gauche du cluster pour lequel vous souhaitez modifier les paramètres de sauvegarde, dans la colonne Identifiant du cluster.
4. Choisissez Actions, puis Modifier.

5. Dans la section Backup, modifiez les champs en fonction de vos besoins en matière de sauvegarde.



   a. Période de conservation des sauvegardes : dans la liste, choisissez le nombre de jours pendant lesquels vous pouvez conserver les sauvegardes automatiques de ce cluster avant de les supprimer.

   b. Fenêtre de sauvegarde : définissez l'heure et la durée quotidiennes pendant lesquelles Amazon DocumentDB doit effectuer des sauvegardes de ce cluster.

      i. Choisissez Sélectionner une fenêtre si vous souhaitez configurer l'heure et la durée de création des sauvegardes.

         Heure de début : dans la première liste, choisissez l'heure de début (UTC) pour démarrer vos sauvegardes automatiques. Dans la deuxième liste, choisissez la minute de l'heure à laquelle vous voulez que les sauvegardes automatiques commencent.

         Durée : dans la liste, choisissez le nombre d'heures à allouer à la création de sauvegardes automatiques.

      ii. Choisissez Aucune préférence si vous souhaitez qu'Amazon DocumentDB choisisse l'heure et la durée de création des sauvegardes.

6. Choisissez Modifier le cluster lorsque vous avez terminé.

Using the AWS CLI

Pour définir une sauvegarde automatique pour un nouvel instantané du cluster Elastic à l'aide de AWS CLI, utilisez l'`create-cluster-snapshot`opération avec les paramètres suivants :

- **`--preferred-backup-window`**—Facultatif. Période quotidienne préférée pendant laquelle les sauvegardes automatisées sont créées. La valeur par défaut est une fenêtre de 30 minutes sélectionnée au hasard dans un intervalle de 8 heures pour chacune d'entre elles. Région AWS

Contraintes :

- Doit être au format `hh24:mi-hh24:mi`.
- Doit être exprimée en heure UTC (Universal Coordinated Time).
- Ne doit pas être en conflit avec la fenêtre de maintenance préférée.
- Doit être de 30 minutes minimum.

- **`--backup-retention-period`**—Facultatif. Nombre de jours de conservation des sauvegardes automatiques. La valeur par défaut est 1.

  Contraintes :

  - Vous devez spécifier une valeur minimale de 1.
  - La plage est comprise entre 1 et 35.

> **Note**
>
> Les sauvegardes automatisées ne sont effectuées que lorsque le cluster est dans un état « actif ».

> **Note**
>
> Vous pouvez également modifier les `backup-retention-period` paramètres `preferred-backup-window` et d'un cluster élastique existant à l'aide de la `aws docdb-elastic update-cluster` commande.

Dans les exemples suivants, remplacez chaque *`espace réservé pour l'entrée utilisateur`* par vos propres informations.

`L'create-clusterexemple suivant crée l'exemple de cluster élastique Amazon DocumentDB avec une période de rétention de 7 jours pour les sauvegardes automatiques et une fenêtre de sauvegarde préférée comprise entre 18 h 00 et 18 h 30 UTC.`

Pour Linux, macOS ou Unix :

```
aws docdb-elastic create-cluster \
```

```
    --cluster-name sample-cluster \
    --shard-capacity 2 \
    --shard-count 2 \
    --admin-user-name SampleAdmin \
    --auth-type PLAIN_TEXT \
    --admin-user-password SamplePass123! \
    --preferred-backup-window 18:00-18:30 \
    --backup-retention-period 7
```

Pour Windows :

```
aws docdb-elastic create-cluster ^
    --cluster-name sample-cluster ^
    --shard-capacity 2 ^
    --shard-count 2 ^
    --admin-user-name SampleAdmin ^
    --auth-type PLAIN_TEXT ^
    --admin-user-password SamplePass123! ^
    --preferred-backup-window 18:00-18:30 ^
    --backup-retention-period 7
```

Arrêt et démarrage d'un cluster élastique Amazon DocumentDB

L'arrêt et le démarrage des clusters élastiques Amazon DocumentDB peuvent vous aider à gérer les coûts des environnements de développement et de test. Au lieu de créer et de supprimer des clusters élastiques chaque fois que vous utilisez Amazon DocumentDB, vous pouvez arrêter temporairement votre cluster lorsqu'il n'est pas nécessaire. Vous pourrez ensuite le recommencer lorsque vous reprendrez vos tests.

Rubriques

Présentation de l'arrêt et du démarrage d'un cluster élastique

Pendant les périodes où vous n'avez pas besoin d'un cluster élastique Amazon DocumentDB, vous pouvez arrêter le cluster. Vous pouvez ensuite à tout moment redémarrer le cluster dès que vous avez besoin de l'utiliser. Le démarrage et l'arrêt simplifient les processus de configuration et de démontage des clusters élastiques utilisés pour le développement, les tests ou des activités similaires

ne nécessitant pas de disponibilité continue. Vous pouvez arrêter et démarrer un cluster élastique en utilisant le AWS Management Console ou le AWS CLI en une seule action.

Lorsque votre cluster élastique est arrêté, le volume de stockage du cluster reste inchangé. Vous êtes facturé uniquement pour le stockage, les instantanés manuels et le stockage des sauvegardes automatiques pendant la fenêtre de conservation spécifiée. Amazon DocumentDB démarre automatiquement votre cluster élastique au bout de sept jours afin qu'il ne prenne aucun retard par rapport aux mises à jour de maintenance requises. Lorsque votre cluster démarrera au bout de sept jours, l'utilisation du cluster élastique recommencera à vous être facturée. Lorsque votre cluster est arrêté, vous ne pouvez pas interroger votre volume de stockage car l'interrogation nécessite que le cluster soit dans l'état disponible.

Lorsqu'un cluster élastique Amazon DocumentDB est arrêté, il ne peut en aucun cas être modifié. Cela inclut la suppression du cluster.

Using the AWS Management Console

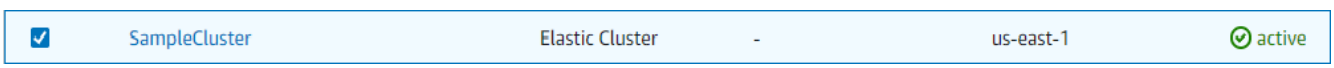
La procédure suivante explique comment arrêter un cluster élastique dans l'état disponible ou démarrer un cluster élastique arrêté.

Pour arrêter ou démarrer un cluster élastique Amazon DocumentDB

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.

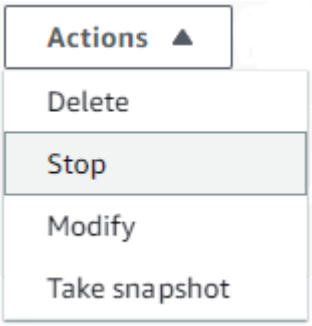
   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans la liste des clusters, choisissez le bouton sur la gauche du nom du cluster que vous voulez arrêter ou démarrer.

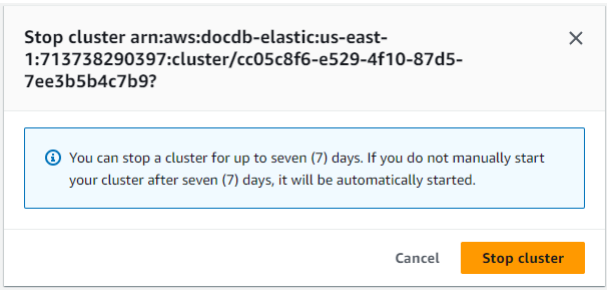
   

4. Choisissez Actions, puis l'action que vous souhaitez exécuter sur le cluster.
   - Si vous souhaitez arrêter le cluster et que celui-ci est disponible :

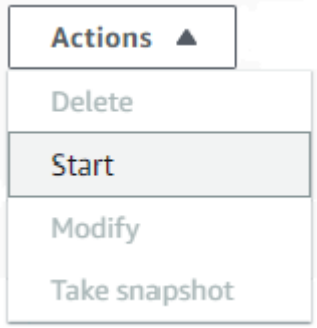
a. Choisissez Arrêter.



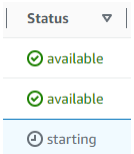
b. Dans la boîte de dialogue de confirmation, confirmez que vous souhaitez arrêter le cluster élastique en choisissant Arrêter le cluster, ou pour maintenir le cluster en cours d'exécution, choisissez Annuler.



- Si vous souhaitez démarrer un cluster et que celui-ci est arrêté, choisissez Start (Démarrer).



5. Surveillez l'état du cluster élastique. Si vous avez démarré le cluster, vous pouvez recommencer à l'utiliser lorsqu'il sera disponible. Pour de plus amples informations, veuillez consulter Déterminer le statut d'un cluster.



Using the AWS CLI

Les exemples de code suivants vous montrent comment arrêter un cluster élastique dans l'état actif ou disponible, ou démarrer un cluster élastique arrêté.

Pour arrêter un cluster élastique à l'aide de AWS CLI, utilisez l'`stop-cluster`opération. Pour démarrer un cluster arrêté, utilisez l'opération `start-cluster`. Les deux opérations utilisent le paramètre `--cluster-arn`.

Paramètre :

- **`--cluster-arn`** : obligatoire. Identifiant ARN du cluster élastique que vous souhaitez arrêter ou démarrer.

Example — Pour arrêter un cluster élastique à l'aide du AWS CLI

Dans les exemples suivants, remplacez chaque *`espace réservé pour l'entrée utilisateur`* par vos propres informations.

Le code suivant arrête le cluster élastique avec un ARN de`arn:aws:docdb-elastic:us-east-1:477568257630:cluster/b9f1d489-6c3e-4764-bb42-da62ceb7bda2`.

> **Note**
>
> Le cluster élastique doit être actif ou disponible.

Pour Linux, macOS ou Unix :

```
aws docdb-elastic stop-cluster \
   --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Pour Windows :

```
aws docdb-elastic stop-cluster ^
   --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Example — Pour démarrer un cluster élastique à l'aide du AWS CLI

Dans les exemples suivants, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations.

Le code suivant démarre le cluster élastique avec un ARN de`arn:aws:docdb-elastic:us-east-1:477568257630:cluster/b9f1d489-6c3e-4764-bb42-da62ceb7bda2`.

> **Note**
>
> Le cluster élastique doit actuellement être arrêté.

Pour Linux, macOS ou Unix :

```
aws docdb-elastic start-cluster \
   --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Pour Windows :

```
aws docdb-elastic start-cluster ^
   --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Opérations que vous pouvez effectuer sur un cluster élastique arrêté

Vous ne pouvez pas modifier la configuration d'un cluster élastique Amazon DocumentDB lorsque celui-ci est arrêté. Vous devez démarrer le cluster avant d'effectuer des opérations d'administration de ce type.

Amazon DocumentDB applique toute maintenance planifiée à votre cluster élastique arrêté uniquement après son redémarrage. Au bout de sept jours, Amazon DocumentDB démarre automatiquement un cluster élastique arrêté afin qu'il ne prenne pas trop de retard dans son état de maintenance. Lorsque le cluster élastique redémarrera, les partitions du cluster recommenceront à vous être facturées.

Lorsqu'un cluster élastique est arrêté, Amazon DocumentDB n'effectue aucune sauvegarde automatique et ne prolonge pas la période de conservation des sauvegardes.

Chiffrement des données au repos pour les clusters Amazon DocumentDB Elastic Cluster DB.

Les rubriques suivantes vous aident à découvrir, à créer et à surveiller les clés de AWS Key Management Service chiffrement pour les clusters élastiques Amazon DocumentDB :

Rubriques

Les clusters élastiques Amazon DocumentDB s'intègrent automatiquement à AWS Key Management Service (AWS KMS) pour la gestion des clés et utilisent une méthode connue sous le nom de cryptage d'enveloppe pour protéger vos données. Pour plus d'informations sur le chiffrement d'enveloppe, consultez Chiffrement d'enveloppe dans le Guide du développeurAWS Key Management Service.

Une AWS KMS key est une représentation logique d'une clé. La clé KMS inclut des métadonnées, telles que l'ID de clé, la date de création, la description et l'état de la clé. La clé KMS contient également les éléments de clé utilisés pour chiffrer et déchiffrer les données. Pour plus d'informations sur les clés KMS, consultez AWS KMS keys dans le Guide du développeur AWS Key Management Service.

Les clusters élastiques Amazon DocumentDB prennent en charge le chiffrement à l'aide de deux types de clés :

- AWSclés détenues : les clusters élastiques Amazon DocumentDB utilisent ces clés par défaut pour chiffrer automatiquement les données personnelles identifiables. Vous ne pouvez pas afficher, gérer ou utiliser les clés AWS appartenant à votre que vous possédez, ni vérifier leur utilisation. Vous n'avez toutefois aucune action ou modifier aucun programme pour protéger les clés qui chiffrent vos données. Pour plus d'informations, consultez la section AWSrelative aux clés détenues dans le Guide du AWS Key Management Service développeur.
- Clés gérées par le client — Symétriques AWS KMS keys que vous créez, possédez et gérez. Comme vous avez le contrôle total de cette couche de chiffrement, vous pouvez effectuer des tâches telles que :

- Établir et maintenir des politiques clés
- Établir et maintenir des politiques et des subventions IAM
- Activation et désactivation désactivation désactivation activation et désactivation
- Rotation des éléments de clé clé éléments clé éléments clé
- Ajout de balises
- Création d'alias clés
- Planification des clés pour la suppression

Pour plus d'informations, consultez la section Clés gérées par le client dans le Guide du AWS Key Management Service développeur.

Important

Vous devez utiliser une clé KMS de chiffrement pour chiffrer votre cluster car Amazon DocumentDB prend uniquement en charge les clés KMS de chiffrement symétrique. N'utilisez pas une clé KMS asymétrique pour tenter de chiffrer les données dans vos clusters Amazon DocumentDB Elastic. Pour plus d'informations, consultez la section Clés asymétriques AWS KMS du Guide du AWS Key Management Service développeur.

Si Amazon DocumentDB ne peut plus accéder à la clé de chiffrement pour un cluster — par exemple, lorsque l'accès à une clé est révoqué — le cluster chiffré est placée dans un état de mise hors service. Dans ce cas, vous pouvez uniquement restaurer le cluster à partir d'une sauvegarde. Pour Amazon DocumentDB, les sauvegardes sont toujours activées pendant 1 jour. En outre, si vous désactivez la clé d'un cluster Amazon DocumentDB chiffré, vous finirez par perdre l'accès en lecture et en écriture à ce cluster. Lorsqu'Amazon DocumentDB rencontre un cluster chiffré par une clé à laquelle il n'a pas accès, il met le cluster en état de terminal. Dans cet état, le cluster n'est plus disponible et l'état actuel de la base de données ne peut pas être récupéré. Pour restaurer le cluster, vous devez réactiver l'accès à la clé de chiffrement pour Amazon DocumentDB, puis restaurer l'accès à la sauvegarde.

Important

Vous ne pouvez pas modifier la clé KMS pour un cluster chiffré une fois que vous l'avez déjà créé. Vous devez prendre soin de déterminer vos besoins en termes de clés de chiffrement avant de créer votre cluster élastique chiffrées.

Comment les clusters élastiques Amazon DocumentDB utilisent les subventions dans AWS KMS

Les clusters Amazon DocumentDB Elastic nécessitent une subvention pour utiliser votre clé gérée par le client.

Lorsque vous créez un cluster chiffré avec une clé gérée par le client, les clusters Amazon DocumentDB Elastic créent une subvention en votre nom en envoyant une `CreateGrant` requête àAWS KMS. Les octrois dans AWS KMS sont utilisés pour accorder aux clusters Amazon DocumentDB Elastic l'accès à une clé KMS dans un compte client.

Les clusters Amazon DocumentDB Elastic nécessitent une autorisation pour utiliser votre clé gérée par le client pour les opérations internes suivantes :

- Envoyez `DescribeKey` des demandes AWS KMS pour vérifier que l'ID de clé KMS symétrique géré par le client, saisi lors de la création d'une collection de suivi ou de géofence, est valide.
- Envoyez `GenerateDataKey` des demandes AWS KMS à pour générer des clés de données cryptées par votre clé gérée par le client.
- Envoyez `Decrypt` des requêtes AWS KMS à pour déchiffrer les clés de données chiffrées afin qu'elles puissent être utilisées pour chiffrer vos données.
- Vous pouvez révoquer l'accès à l'octroi ou supprimer l'accès du service à la clé gérée par le client à tout moment. Dans ce cas, les clusters Amazon DocumentDB Elastic ne pourront accéder à aucune des données chiffrées par la clé gérée par le client, ce qui affecte les opérations qui dépendent de ces données.

Création d'une clé gérée par le client éléments de clé

Vous pouvez créer une clé symétrique gérée par le client à l'aide de la AWS Management Console ou de l'AWS KMSAPI.

Création de clés symétriques gérées par le client

Suivez les étapes relatives à la création d'une clé symétrique gérée par le client dans le Guide du AWS Key Management Service développeur.

Politique de clé

Les politiques de clés contrôlent l'accès à votre clé gérée par le client. Chaque clé gérée par le client doit avoir exactement une politique de clé, qui contient des instructions qui déterminent les personnes

pouvant utiliser la clé et comment elles peuvent l'utiliser. Lorsque vous créez votre clé gérée par le client, vous pouvez spécifier une politique de clé. Pour plus d'informations, consultez les informations relatives à l'accès par clé KMS dans la AWS Key Management Servicevue d'ensemble du Guide du AWS Key Management Service développeur.

Pour utiliser votre clé gérée par le client avec les ressources du cluster Amazon DocumentDB, les opérations API suivantes doivent être autorisées dans la politique de clés :

- `kms:CreateGrant`— Ajoute une subvention à une clé gérée par le client. Les autorisations contrôlent l'accès à une clé KMS spécifiée, qui permet d'accéder aux opérations d'autorisation requises par Amazon Location Service. Pour plus d'informations sur l'utilisation des subventions, consultez la section Subventions AWS KMS dans le Guide du AWS Key Management Service développeur.
- `kms:DescribeKey`— Fournit les détails de la clé gérée par le client pour permettre à Docdb Elastic de valider la clé.
- `kms:Decrypt`— Permet à Docdb Elastic d'utiliser la clé de données cryptée stockée pour accéder aux données cryptées.
- `kms:GenerateDataKey`— Permet à Docdb Elastic de générer une clé de données cryptée et de la stocker car la clé de données n'est pas immédiatement utilisée pour chiffrer.

Pour plus d'informations, consultez les sections Autorisations pour les AWS services dans les politiques clés et Résolution des problèmes d'accès par clé dans le Guide du AWS Key Management Service développeur.

Restreindre l'accès aux clés gérées par le client via des politiques IAM

Outre les politiques de clé KMS, vous pouvez également restreindre les autorisations de clé KMS dans une politique IAM.

Vous pouvez renforcer la politique IAM de différentes manières. Par exemple, pour limiter l'utilisation de la la clé gérée par le client aux seules requêtes provenant des clusters Amazon DocumentDB Elastic, vous pouvez utiliser la valeur la clé de `kms:ViaService` condition avec la `docdb-elastic.<region-name>.amazonaws.com` valeur.

Pour plus d'informations, consultez Autoriser des utilisateurs d'autres comptes à utiliser une clé KMS dans le Guide du développeur AWS Key Management Service.

Surveillance de vos clés de chiffrement pour les clusters Amazon DocumentDB Elastic Cluster DB.

Lorsque vous utilisez une clé gérée par le AWS KMS key client avec vos ressources Docdb Elastic, vous pouvez utiliser AWS CloudTrail ou Amazon CloudWatch Logs pour suivre les demandes envoyées par Docdb Elastic. AWS KMS

Les exemples suivants concernent AWS CloudTrail des événements destinés à `CreateGrant` `GenerateDataKeyWithoutPlainTextDecrypt`, et `DescribeKey` à surveiller des AWS KMS key opérations appelées par les clusters élastiques Amazon DocumentDB pour accéder à des données cryptées par votre clé gérée par le client :

CreateGrant


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-05-09T23:04:20Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "docdb-elastic.amazonaws.com"
    },
    "eventTime": "2023-05-09T23:55:48Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-east-1",
```

```
    "sourceIPAddress": "docdb-elastic.amazonaws.com",
    "userAgent": "docdb-elastic.amazonaws.com",
    "requestParameters": {
        "retiringPrincipal": "docdb-elastic.us-east-1.amazonaws.com",
        "granteePrincipal": "docdb-elastic.us-east-1.amazonaws.com",
        "operations": [
            "Decrypt",
            "Encrypt",
            "GenerateDataKey",
            "GenerateDataKeyWithoutPlaintext",
            "ReEncryptFrom",
            "ReEncryptTo",
            "CreateGrant",
            "RetireGrant",
            "DescribeKey"
        ],
        "keyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

GenerateDataKey


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-05-10T18:02:59Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "docdb-elastic.amazonaws.com"
    },
    "eventTime": "2023-05-10T18:03:25Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "docdb-elastic.amazonaws.com",
    "userAgent": "docdb-elastic.amazonaws.com",
    "requestParameters": {
        "keySpec": "AES_256",
        "keyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
```

```
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Decrypt

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-05-10T18:05:49Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "docdb-elastic.amazonaws.com"
    },
    "eventTime": "2023-05-10T18:06:19Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "docdb-elastic.amazonaws.com",
    "userAgent": "docdb-elastic.amazonaws.com",
```

```
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


DescribeKey


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-05-09T23:04:20Z",
                "mfaAuthenticated": "false"
```

```
            }
        },
        "invokedBy": "docdb-elastic.amazonaws.com"
    },
    "eventTime": "2023-05-09T23:55:48Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "docdb-elastic.amazonaws.com",
    "userAgent": "docdb-elastic.amazonaws.com",
    "requestParameters": {
        "keyId": "alias/SampleKmsKey"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

En savoir plus

Les ressources suivantes fournissent plus d'informations sur le chiffrement des données au repos :

- Pour plus d'informations sur AWS KMS les concepts, consultez les concepts AWS Key Management Service de base dans le Guide du AWS Key Management Service développeur.
- Pour plus d'informations sur AWS KMS la sécurité, consultez les bonnes pratiques AWS Key Management Service de sécurité du Guide du AWS Key Management Service développeur.

Rôles liés aux services dans les clusters élastiques

Les clusters élastiques Amazon DocumentDB utilisent des rôles liés à un AWS Identity and Access Management service (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié aux clusters élastiques Amazon DocumentDB. Les rôles liés aux services sont prédéfinis par les clusters élastiques Amazon DocumentDB et incluent toutes les autorisations requises par le service pour appeler AWS d'autres services en votre nom.

Un rôle lié à un service facilite l'utilisation des clusters élastiques Amazon DocumentDB, car vous n'avez pas à ajouter manuellement les autorisations nécessaires. Les clusters élastiques Amazon DocumentDB définissent les autorisations associées à ses rôles liés aux services et, sauf indication contraire, seuls les clusters élastiques Amazon DocumentDB peuvent assumer ces rôles. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM. Vous pouvez supprimer les rôles uniquement après la suppression préalable de leurs ressources connexes. Cela protège les ressources de vos clusters élastiques Amazon DocumentDB, car vous ne pouvez pas supprimer par inadvertance l'autorisation d'accès aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés à un service, consultez la section AWS Services qui fonctionnent avec IAM et recherchez les services marqués d'un Oui dans la colonne Rôle lié au service. Choisissez un Oui ayant un lien permettant de consulter les détails du rôle pour ce service.

Autorisations de rôle liées à un service pour les clusters élastiques

Les clusters élastiques Amazon DocumentDB utilisent le rôle lié à un service nommé pour permettre aux clusters élastiques `AWS ServiceRoleForDocDB-Elastic` Amazon DocumentDB d'appeler des AWS services au nom de vos clusters.

Ce rôle lié à un service est associé à une politique appelée `AmazonDocDB-ElasticServiceRolePolicy` qui lui accorde l'autorisation d'opérer dans votre compte. La politique d'autorisation des rôles permet aux clusters élastiques Amazon DocumentDB d'effectuer les actions suivantes sur les ressources spécifiées :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "AWS/DocDB-Elastic"
                    ]
                }
            }
        }
    ]
}
```

> **Note**
>
> 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. Si le message d'erreur suivant s'affiche : « Impossible de créer la ressource. Vérifiez que vous détenez l'autorisation de créer un rôle lié au service. Sinon, attendez et réessayez plus tard. » , assurez-vous que les autorisations suivantes sont activées :

```
{
"Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/docdb-elastic.amazonaws.com/
AWSServiceRoleForDocDB-Elastic",
    "Condition": {
"StringLike": {
"iam:AWSServiceName":"docdb-elastic.amazonaws.com"
        }
    }
}
```

Pour plus d'informations, consultez la section Autorisations relatives aux rôles liés à un service dans le guide de l'utilisateur AWS d'Identity and Access Management.

Création d'un rôle lié à un service pour les clusters élastiques Amazon DocumentDB

Vous n’avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous créez une instance de base de données, les clusters élastiques Amazon DocumentDB créent pour vous le rôle lié au service.

Modification d'un rôle lié à un service pour les clusters élastiques Amazon DocumentDB

Les clusters élastiques Amazon DocumentDB ne vous permettent pas de modifier le rôle lié à un `AWS ServiceRoleForDocDB-Elastic` service. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas changer le nom du rôle, car plusieurs entités peuvent faire référence à ce rôle. Néanmoins, vous pouvez modifier la description du rôle à l’aide d’IAM. Pour plus d'informations, consultez la section Modification d'un rôle lié à un service dans le guide de l'utilisateur AWS d'Identity and Access Management.

Suppression d'un rôle lié à un service pour les clusters élastiques Amazon DocumentDB

Si vous n’avez plus besoin d’utiliser une fonction 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 supprimer tous les clusters avant de pouvoir supprimer le rôle lié à un service.

Nettoyage d'un rôle lié à un service

Avant de pouvoir utiliser IAM pour supprimer un rôle lié à un service, vous devez d'abord vérifier qu'aucune session n'est active pour le rôle et supprimer toutes les ressources utilisées par le rôle.

Pour vérifier si le rôle lié à un service possède une session active dans la console IAM :

1. Connectez-vous à AWS Management Console et ouvrez la console IAM.
2. Dans le panneau de navigation de la console IAM, sélectionnez Roles (Rôles). Ensuite, sélectionnez le nom (et non la case à cocher) du rôle `AWS ServiceRoleForDocDB-Elastic`.
3. Sur la page Summary (Récapitulatif) du rôle sélectionné, choisissez l'onglet Access Advisor.

> **Note**
>
> Si vous ne savez pas si Amazon DocumentDB Elastic Clusters utilise le `AWS ServiceRoleForDocDB-Elastic` rôle, vous pouvez essayer de le supprimer. Si le service

utilise le rôle, la suppression échoue et vous pouvez voir Régions AWS où le rôle est utilisé. Si le rôle est utilisé, vous devez attendre que la session se termine avant de pouvoir le supprimer. Vous ne pouvez pas révoquer la session d'un rôle lié à un service.
Si vous souhaitez supprimer le `AWS ServiceRoleForDocDB-Elastic` rôle, vous devez d'abord supprimer tous vos clusters.

Suppression de tous vos clusters

Pour supprimer un cluster dans la console Amazon DocumentDB :

1. Connectez-vous à la console Amazon DocumentDB AWS Management Consoleet ouvrez-la.
2. Dans le panneau de navigation, choisissez Clusters.
3. Choisissez le cluster que vous souhaitez supprimer.
4. Pour Actions, choisissez Supprimer.
5. Si vous êtes invité à créer un instantané final ? , choisissez Oui ou Non.
6. Si vous avez choisi Oui à l'étape précédente, dans le champ Nom de l'instantané final, saisissez le nom de votre instantané final.
7. Sélectionnez Delete.

> **Note**
>
> Vous pouvez utiliser la console IAM, l'interface de ligne de commande IAM ou l'API IAM pour supprimer le rôle lié à un service `AWS ServiceRoleForDocDB-Elastic`. Pour plus d'informations, consultez la section Suppression d'un rôle lié à un service dans le guide de l'utilisateur AWS d'Identity and Access Management.

Surveillance Amazon DocumentDB

La surveillance de vosAWS services est un élément important pour maintenir la santé et le fonctionnement optimal de vos systèmes. Il est judicieux de recueillir les données de surveillance de toutes les parties de votreAWS solution afin de pouvoir déboguer et corriger les éventuelles défaillances ou dégradations. Avant de commencer à surveiller vosAWS solutions, nous vous recommandons de réfléchir aux questions suivantes et de formuler des réponses :

- Quels sont les objectifs de la surveillance ?
- Quelles ressources allez-vous surveiller ?
- Selon quelle fréquence allez-vous surveiller ces ressources ?
- Quels outils de surveillance utiliser ?
- Qui est responsable d'effectuer la surveillance ?
- Qui doit être informé et par quels moyens en cas de problème ?

Pour comprendre vos modèles de performances actuelles, identifier les anomalies de rendement et formuler des méthodes pour régler les problèmes, vous devriez établir des mesures des performances de base pour diverses périodes et dans différentes conditions de charge. Lorsque vous surveillez votreAWS solution, nous vous recommandons de stocker vos données de surveillance historiques à des fins de référence future et pour établir vos bases de référence.

En général, les valeurs acceptables pour les métriques de performances dépendent de vos données de référence et de l'activité de votre application. Enquêtez sur les écarts cohérents ou tendanciels de vos données de référence. Voici quelques conseils sur les types spécifiques de mesures :

- Forte utilisation de l'UC et de la RAM — Des valeurs importantes de l'utilisation de l'UC et de la RAM peuvent être appropriées, pourvu qu'elles soient attendues et conformes aux objectifs pour votre application (comme le débit ou la simultanéité).
- Utilisation de l'espace de stockage — Enquêtez sur la utilisation de l'espace de stockage (`VolumeBytesUsed`) si l'espace utilisé est constamment égal ou supérieur à 85 pour cent de l'espace total du volume de stockage. Déterminez s'il est possible de supprimer des données du volume de stockage ou d'archiver des données sur un système différent pour libérer de l'espace. Pour plus d'informations, consultez Stockage Amazon DocumentDB et Quotas et limites Amazon DocumentDB.

- Trafic réseau — Pour le trafic réseau, discutez avec votre administrateur pour connaître le débit attendu pour votre domaine réseau et votre connexion Internet. Enquêtez sur le trafic réseau si le débit est constamment inférieur à vos attentes.
- Connexions de la base de données — Envisagez de limiter les connexions de la base de données si vous constatez un nombre important de connexions utilisateur conjointement avec une baisse des performances de l'instance et des temps de réponse. Le bon nombre de connexions utilisateur pour votre instance dépendra de votre classe d'instance et de la complexité des opérations exécutées.
- Métriques IOPS — Les valeurs attendues pour les métriques d'IOPS par seconde dépendent de la spécification du disque et de la configuration du serveur, donc utilisez vos données de référence pour connaître les caractéristiques typiques. Déterminer si les valeurs sont constamment différentes par rapport à vos données de référence. Pour de meilleures performances IOPS, veillez à ce que votre ensemble de travail typique puisse être chargé en mémoire pour minimiser les opérations de lecture et écriture.

Amazon DocumentDB (compatible MongoDB) propose diversesCloudWatch métriques Amazon que vous pouvez surveiller pour connaître l'état et les performances de vos clusters et instances Amazon DocumentDB. Vous pouvez consulter les métriques Amazon DocumentDB à l'aide de divers outils, notamment la console Amazon DocumentDBAWS CLI,CloudWatch l'API et Performance Insights.

Rubriques

Surveillance de l'état d'un cluster Amazon DocumentDB

Le statut d'un cluster indique son état. Vous pouvez afficher le statut d'un cluster à l'aide de la console Amazon DocumentDB ou de laAWS CLI`describe-db-clusters` commande.

Rubriques

Valeurs de statut de cluster

Le tableau suivant répertorie les valeurs valides pour le statut d'un cluster.

Statut du cluster	Description
`active`	Le cluster est actif. Ce statut s'applique uniquement aux clusters élastiques.
`available`	Le cluster est sain et disponibl e. Cet état ne s'applique qu'aux clusters basés sur des instances.
`backing-up`	Le cluster est en cours de sauvegarde.
`creating`	Le cluster est en cours de création. Il n'est pas accessible pendant sa création.
`deleting`	Le cluster est en cours de suppression. Il n'est pas accessible pendant sa suppression.
`failing-over`	Un basculement à partir de l'instance principale vers un réplica Amazon DocumentDB est en cours.

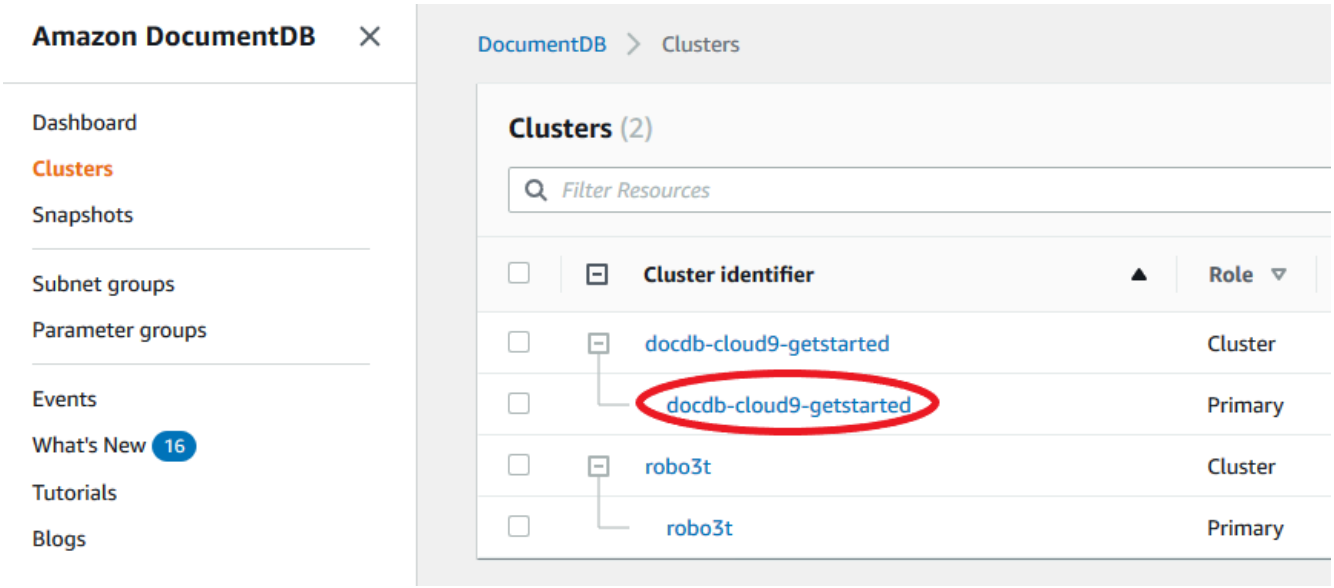
Statut du cluster	Description
`inaccessible-encryption-credentials`	La clé AWS KMS utilisée pour chiffrer ou déchiffrer le cluster n'est pas accessible.
`maintenance`	Une mise à jour de maintenance est appliquée au cluster. Cet état est utilisé pour la maintenance de niveau de cluster qu'Amazon DocumentDB planifie suffisamment à l'avance.
`migrating`	Un instantané de cluster est en cours de restauration à partir d'un cluster.
`migration-failed`	Échec d'une migration.
`modifying`	Le cluster est en cours de modification en raison d'une demande du client de modification du cluster.
`renaming`	Le cluster est actuellement renommé en raison d'une demande du client pour le renommer.
`resetting-master-credentials`	Les informations d'identification principales du cluster sont en cours de réinitialisation, en raison d'une demande du client pour les réinitialiser.
`upgrading`	La version du moteur du cluster est en cours de mise à niveau.

Surveillance de l'état d'un cluster

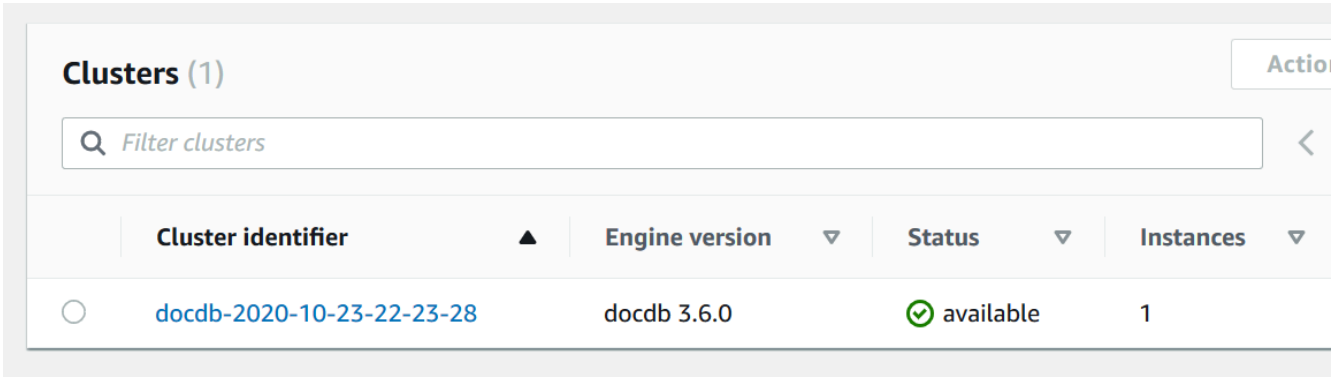
Using the AWS Management Console

Lorsque vous utilisez la AWS Management Console pour déterminer le statut d'un cluster, utilisez la procédure suivante.

1. Connectez-vous à laAWS Management Console et ouvrez la console Amazon DocumentDB à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.
3. Dans la zone de navigation Clusters, vous verrez la colonne Identifiant du cluster. Vos instances sont répertoriées sous clusters, comme sur la capture d'écran ci-dessous.



4. Dans la colonne Identifiant du cluster, recherchez le nom de l'instance qui vous intéresse. Ensuite, pour connaître l'état de l'instance, parcourez cette ligne jusqu'à la colonne Status, comme indiqué ci-dessous.



Using the AWS CLI

Lorsque vous utilisez la AWS CLI pour déterminer le statut d'un cluster, utilisez l'opération `describe-db-clusters`. Le code suivant trouve le statut du cluster `sample-cluster`.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster  \
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

Pour Windows :

```
aws docdb describe-db-clusters ^
    --db-cluster-identifier sample-cluster  ^
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

Le résultat de cette opération ressemble à ceci.

```
[
    [
        "sample-cluster",
        "available"
    ]
]
```

Surveillance de l'état d'une instance Amazon DocumentDB

Amazon DocumentDB fournit des informations sur l'état actuel de chaque instance configurée dans la base de données.

Il existe trois types de statuts que vous pouvez consulter pour une instance Amazon DocumentDB :

- État de l'instance : cet état est affiché dans la colonne État de la table du clusterAWS Management Console et indique l'état actuel du cycle de vie de l'instance. Les valeurs affichées dans la colonne Status sont dérivées du`Status` champ de la réponse de l'`DescribeDBCluster`API.
- État de santé de l'instance : cet état est indiqué dans la colonne État de santé de l'instance de la table de cluster duAWS Management Console et indique si le moteur de base de données,

le composant responsable de la gestion et de la récupération des données, est en cours d'exécution. Les valeurs affichées dans la colonne État de santé de l'instance sont basées sur la métriqueCloudWatch`EngineUptime` du système Amazon.

- État de maintenance : cet état est affiché dans la colonne Maintenance du tableau des clusters duAWS Management Console et indique l'état de tout événement de maintenance qui doit être appliqué à une instance. L'état de maintenance est indépendant du statut de l'autre instance et est dérivé de l'`PendingMaintenanceAction`API. Pour plus d'informations sur le statut de maintenance, consultez Maintenance Amazon DocumentDB.

Rubriques

Valeurs de l’état d'instance

Le tableau suivant répertorie les valeurs de statut possibles pour les instances et la façon dont vous êtes facturé pour chaque statut. Il indique si vous serez facturé pour l'instance et le stockage, uniquement pour le stockage, ou si vous ne serez pas facturé. Pour tous les statuts d'instance, vous êtres toujours facturé pour l'utilisation de la sauvegarde.

Statut de l'instance	Facturé	Description
`available`	Facturé	L'instance est saine et disponible.
`backing-up`	Facturé	L'instance est en cours de sauvegarde.
`configuring-log-exports`	Facturé	La publication des fichiersCloudWatch journaux dans Amazon Logs est en cours d'activation ou de désactivation pour cette instance.
`creating`	Non facturé	L'instance est en cours de création. L'instance n'est pas accessible pendant sa création.

Statut de l'instance	Facturé	Description
`deleting`	Non facturé	L'instance est en cours de suppression.
`failed`	Non facturé	L'instance a échoué et Amazon DocumentDB n'a pas pu la récupérer. Pour récupérer les données, effectuez unepoint-in-time restauration jusqu'à l'heure de restauration la plus récente de l'instance.
`inaccessible-encryption-credentials`	Non facturé	La clé AWS KMS utilisée pour chiffrer ou déchiffrer l'instance n'est pas accessible.
`incompatible-network`	Non facturé	Amazon DocumentDB tente d'effectuer une action de récupération sur une instance, mais n'y parvient pas, car l'état du VPC empêche l'exécution de l'action. Cette situation peut se produire si, par exemple, toutes les adresses IP disponibles d'un sous-réseau étaient en cours d'utilisation et qu'Amazon DocumentDB n'a pas pu obtenir d'adresse IP pour l'instance.
`maintenance`	Facturé	Amazon DocumentDB applique une mise à jour de maintenance à l'instance. Cet état est utilisé pour la maintenance de niveau d'instance qu'Amazon DocumentDB planifie suffisamment à l'avance. Nous avons évalué des façons d'exposer davantage d'actions de maintenance aux clients par le biais de cet état.
`modifying`	Facturé	L'instance est en cours de modification en raison d'une demande de modification de l'instance.

Statut de l'instance	Facturé	Description
`rebooting`	Facturé	L'instance est en cours de redémarrage en raison d'une demande ou d'un processus Amazon DocumentDB nécessitant le redémarrage de l'instance.
`renaming`	Facturé	L'instance est en cours d'être renommée en raison d'une demande pour la renommer.
`resetting -master-c redentials`	Facturé	Les informations d'identification principales de l'instance sont en cours de réinitialisation en raison d'une demande pour les réinitial iser.
`restore-error`	Facturé	L'instance a rencontré une erreur lors de la restauration à partir d'un instantanépoint-in-time ou à partir d'un instantané.
`starting`	Facturé pour stockage	L'instance démarre.
`stopped`	Facturé pour stockage	L'instance est arrêtée.
`stopping`	Facturé pour stockage	L'instance est en cours d'arrêt.
`storage-full`	Facturé	L'instance a atteint son attribution de capacité de stockage. Il s'agit d'un état critique, qui doit être corrigé immédiate ment ; augmentez votre stockage en modifiant l'instance. Configurez lesCloudW atch alarmes Amazon de manière à ce qu'elles vous préviennent lorsque l'espace de stockage devient faible, afin d'éviter cette situation.

Surveillance de l'état de l'instance à l'aide deAWS Management Console ouAWS CLI

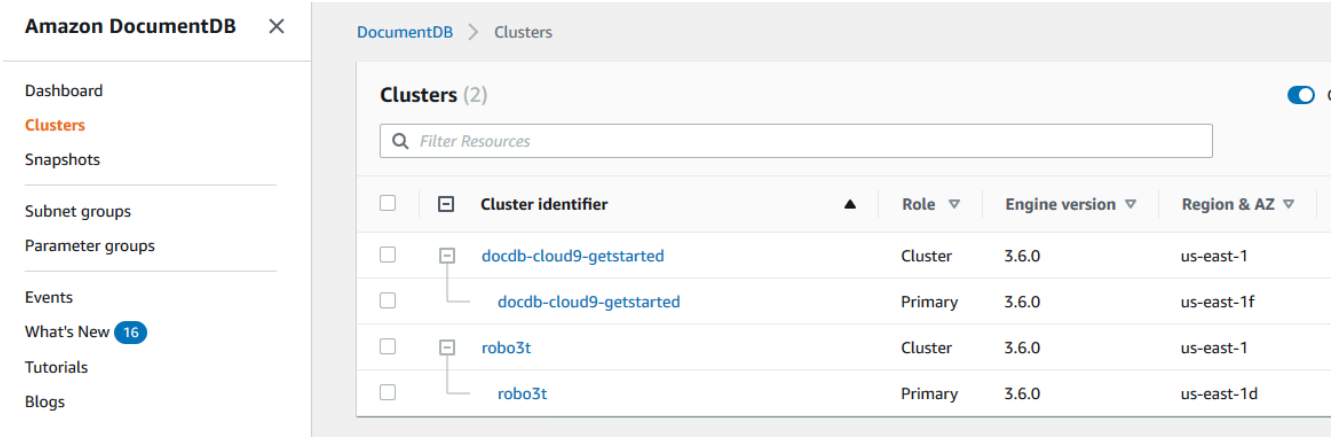
Utilisez leAWS Management Console ouAWS CLI pour surveiller l'état de votre instance.

Using the AWS Management Console

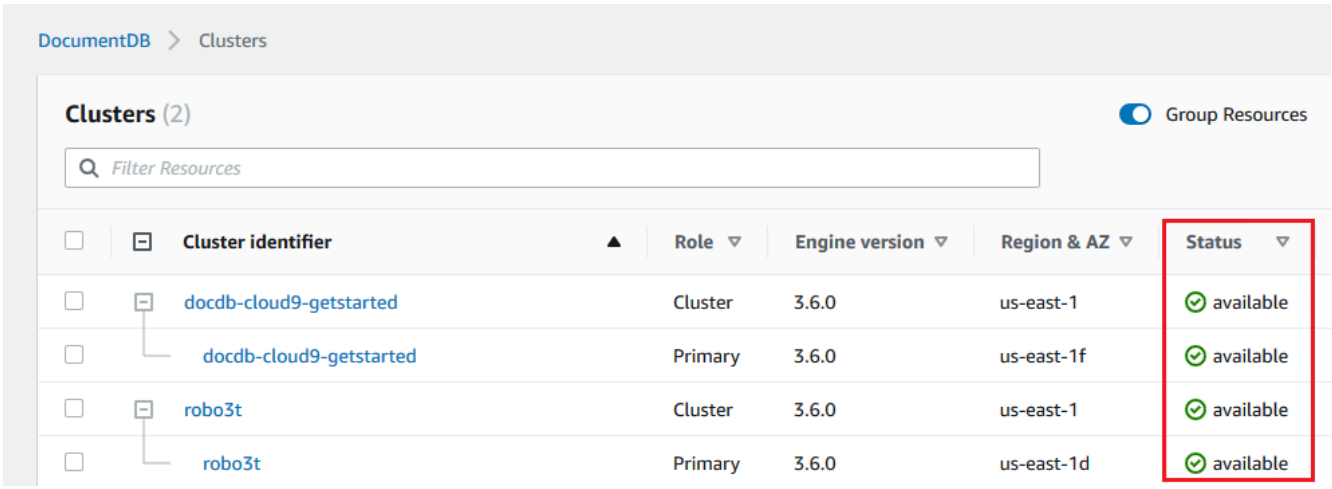
Lorsque vous utilisez la AWS Management Console pour déterminer le statut d'un cluster, utilisez la procédure suivante.

1. Connectez-vous à laAWS Management Console et ouvrez la console Amazon DocumentDB à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.

> **Note**
>
> Notez que dans la zone de navigation Clusters, la colonne Identifiant du cluster affiche à la fois les clusters et les instances. Les instances sont répertoriées sous les clusters, comme dans l'image ci-dessous.



3. Trouvez le nom de l'instance qui vous intéresse. Ensuite, pour trouver le statut de l'instance, lire dans cette ligne à la colonne Statut, comme ci-dessous.



Using the AWS CLI

Lorsque vous utilisez la AWS CLI pour déterminer le statut d'un cluster, utilisez l'opération `describe-db-instances`. Le code suivant trouve le statut de l'instance `sample-cluster-instance-01`.

Pour Linux, macOS ou Unix :

```
aws docdb describe-db-instances \
          --db-instance-identifier sample-cluster-instance-01  \
          --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceStatus]'
```

Pour Windows :

```
aws docdb describe-db-instances ^
          --db-instance-identifier sample-cluster-instance-01  ^
          --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceStatus]'
```

Le résultat de cette opération ressemble à ceci.

```
[
          [
              "sample-cluster-instance-01",
              "available"
          ]
      ]
```

Valeurs de statut d'instance

Le tableau suivant répertorie les valeurs possibles de statut d'instance. La colonne Instance Health, située dans le tableau Clusters duAWS Management Console, indique si le moteur de base de données, le composant responsable du stockage, de la gestion et de la récupération des données, fonctionne normalement. Cette colonne indique également si la métrique du`EngineUptime` système, disponible dansCloudWatch, indique l'état de santé de chaque instance.

État d'une instance	Description
sain	Le moteur de base de données s'exécute dans l'instance Amazon DocumentDB.
mauvais pour la santé	Le moteur de base de données ne fonctionne pas ou a redémarré il y a moins d'une minute.

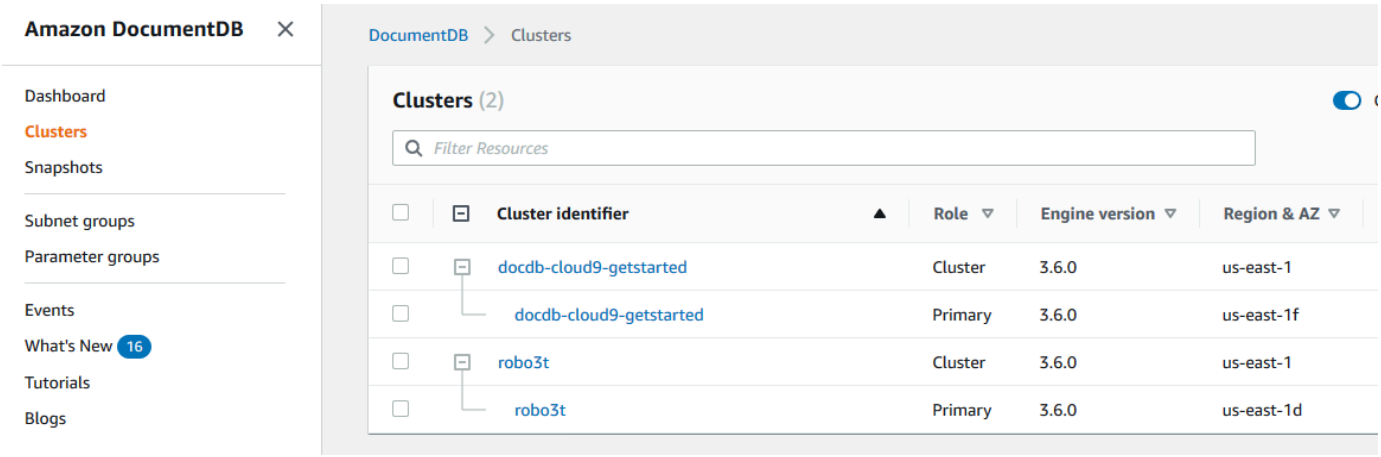
Surveillance de l'état de santé de l'instance à l'aide duAWS Management Console

Utilisez leAWS Management Console pour surveiller l'état de santé de votre instance.

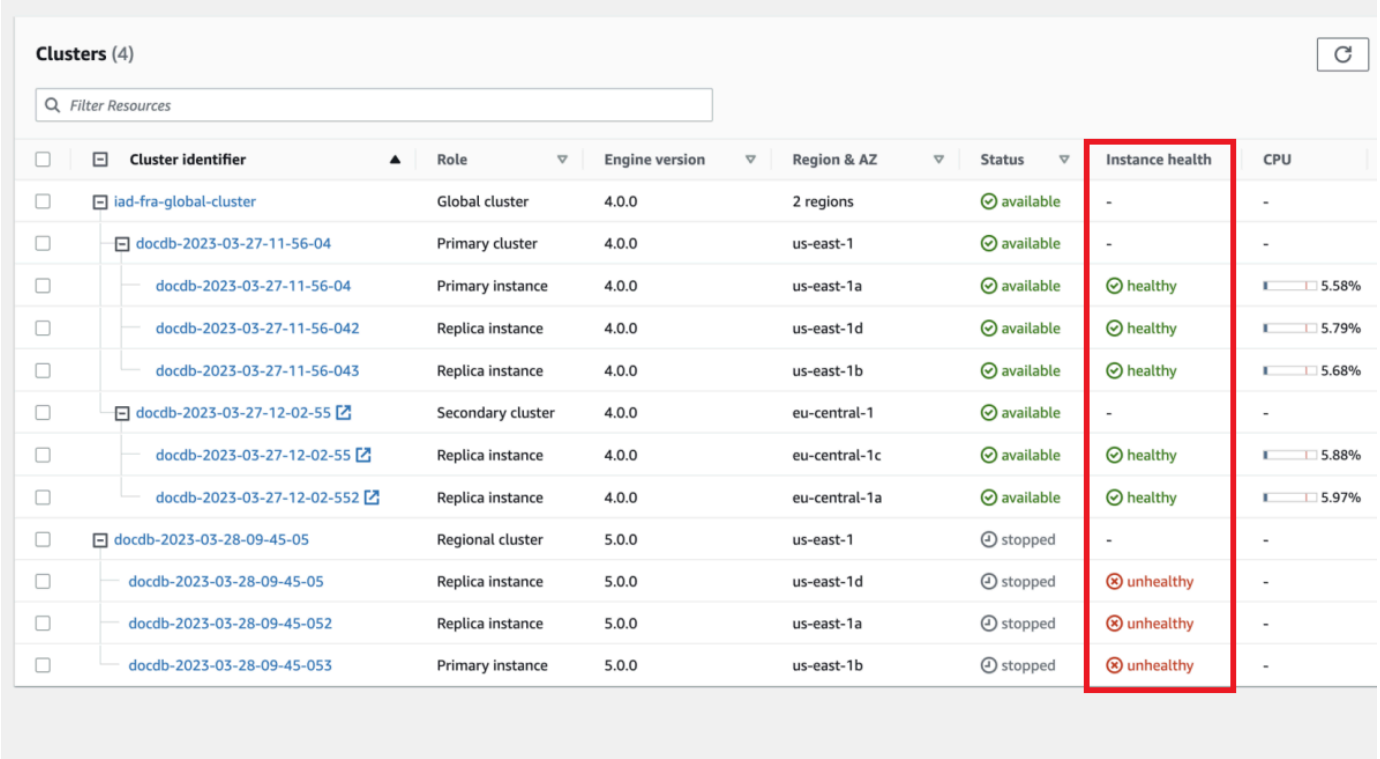
Lors de l'utilisationAWS Management Console, suivez les étapes suivantes pour comprendre l'état de santé de l'instance.

1. Connectez-vous à laAWS Management Console et ouvrez la console Amazon DocumentDB à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.

> **Note**
>
> Dans la zone de navigation Clusters, la colonne Cluster Identifier indique à la fois les clusters et les instances. Les instances sont répertoriées sous les clusters, comme dans l'image ci-dessous.



3. Trouvez le nom de l'instance qui vous intéresse. Ensuite, pour connaître l'état de l'instance, parcourez cette ligne jusqu'à la colonne État de l'instance, comme indiqué dans l'image suivante :



> **Note**
>
> Le sondage sur l'état de santé de l'instance a lieu toutes les 60 secondes et est basé sur la métriqueCloudWatch`EngineUptime` du système. Les valeurs de la colonne État de l'instance sont automatiquement mises à jour.

Affichage des recommandations Amazon DocumentDB

Amazon DocumentDB propose une liste de recommandations automatisées pour les ressources de base de données, telles que les instances et les clusters. Ces recommandations offrent des conseils quand aux bonnes pratiques en analysant vos configurations de cluster et d'instance.

Pour obtenir un exemple de ces recommandations, consultez les exemples suivants :

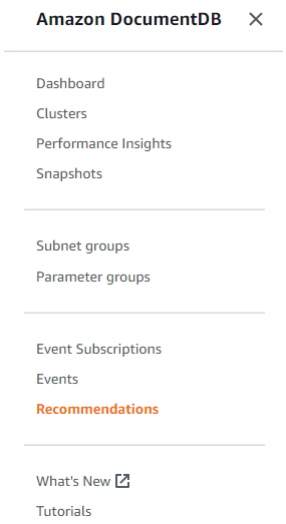
Type	Description	Recommandation	Informations supplémentaires
Une instance	Le cluster ne contient qu'une seule instance	Performances et disponibilité : nous recommandons d'ajouter une autre instance avec la même classe d'instance dans une zone de disponibilité différente.	Haute disponibilité et réplication d'Amazon DocumentDB

Amazon DocumentDB génère des recommandations pour une ressource lors de la création ou de la modification de celle-ci. Amazon DocumentDB analyse également périodiquement vos ressources, et génère des recommandations.

Pour consulter les recommandations d'Amazon DocumentDB et y donner suite

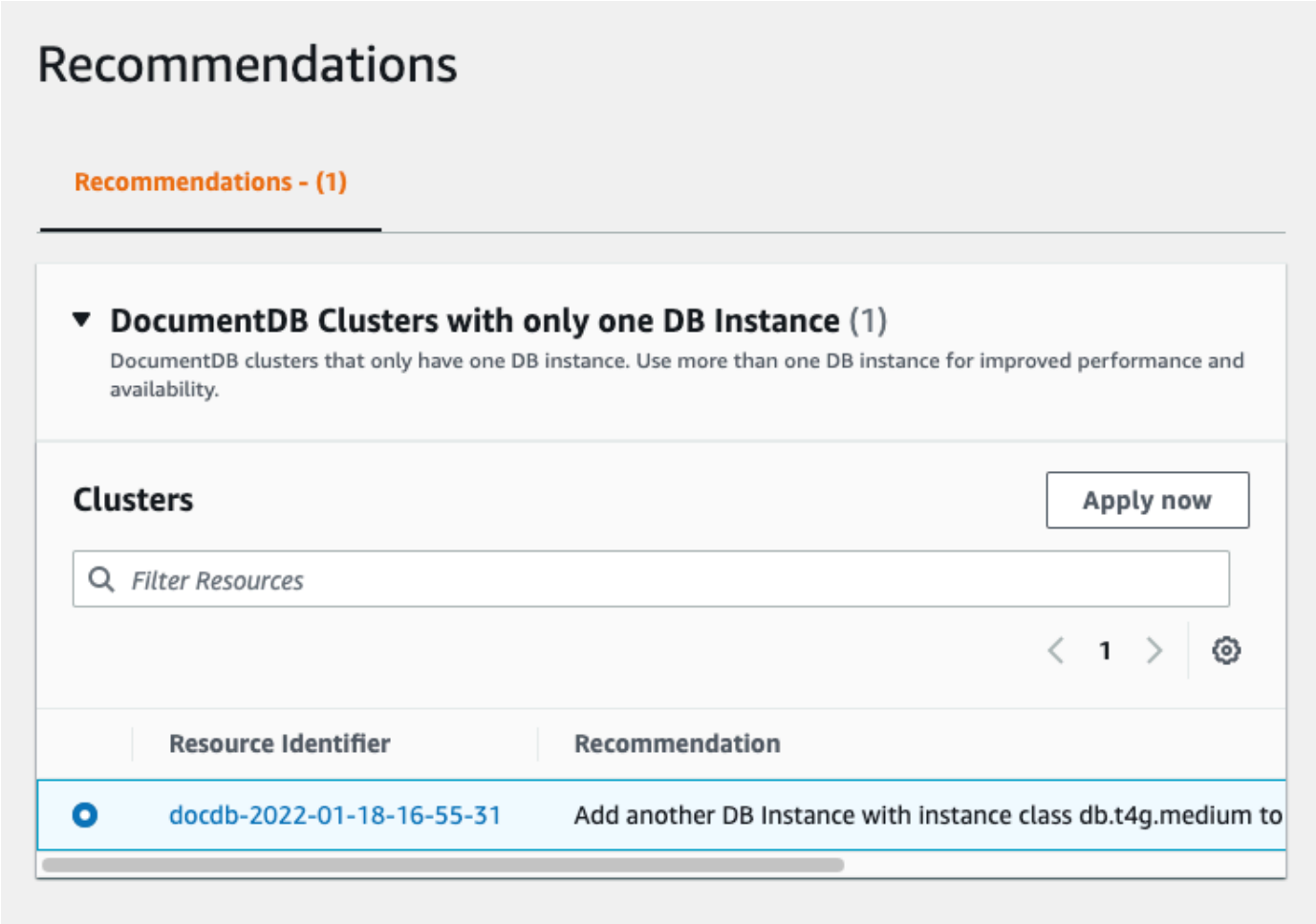
1. Connectez-vous à laAWS Management Console et ouvrez la console Amazon DocumentDB à l'adresse https://console.aws.amazon.com/docdb.

2. Dans le volet de navigation, choisissez Recommandations :



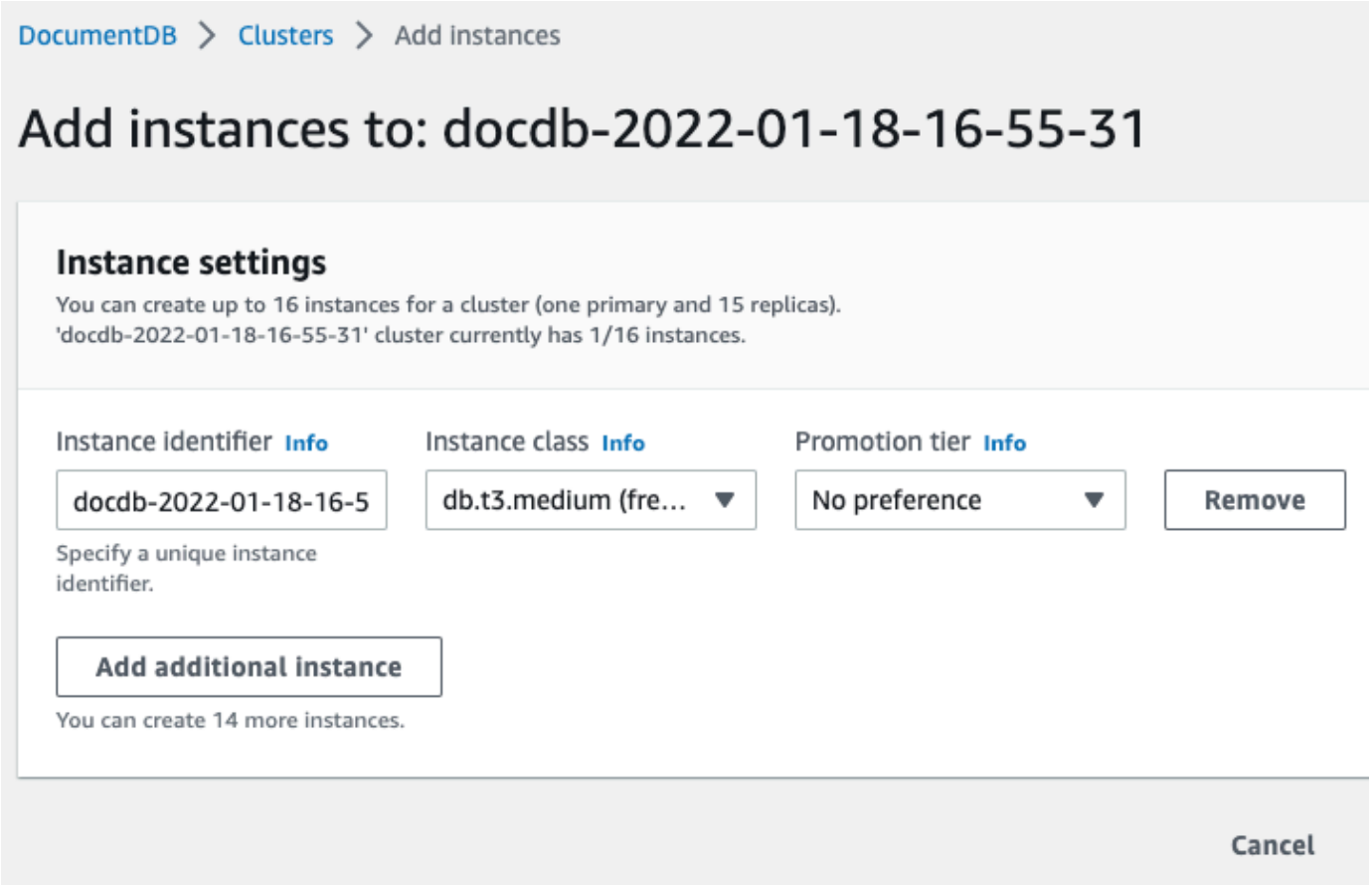
3. Dans la boîte de dialogue Recommandations, développez la section qui vous intéresse et sélectionnez la tâche recommandée.

   Dans l'exemple ci-dessous, la tâche recommandée s'applique à un cluster Amazon DocumentDB comportant une seule instance. Il est recommandé d'ajouter une autre instance pour améliorer les performances et la disponibilité.



4. Cliquez sur Appliquer maintenant.

   Pour cet exemple, la boîte de dialogue Ajouter des instances apparaît :



5. Modifiez les paramètres de votre nouvelle instance et cliquez sur Créer.

Utilisation des abonnements à des événements Amazon DocumentDB

Amazon DocumentDB utilise Amazon Simple Notification Service (Amazon SNS) pour adresser des notifications lorsqu'un événement Amazon DocumentDB se produit. Ces notifications peuvent être sous l'une des formes prises en charge par Amazon SNSRégion AWS, telles qu'un e-mail, un SMS ou un appel à un point de terminaison HTTP.

Amazon DocumentDB regroupe ces événements en catégories auxquelles vous abonner afin d'être informé lorsqu'un événement de cette catégorie se produit. Vous pouvez vous abonner à une catégorie d'événement pour une instance, un cluster, un groupe de base de données ou un groupe de base de données. Par exemple, si vous vous vous abonner à la catégorie de Backup d'une instance donnée, vous recevez une notification chaque fois que survient un événement lié à la sauvegarde et qui affecte l'instance. Vous recevez également une notification en cas de modification d'un abonnement à un événement.

Les événements se produisant au niveau du cluster et de l'instance, vous recevez les événements si vous vous vous vous vous abonner à un cluster ou à une instance.

Les abonnements aux événements sont envoyés aux adresses que vous fournissez lorsque vous créez l'abonnement. Il se peut que vous souhaitiez créer plusieurs abonnements, tels qu'un abonnement recevant toutes les notifications d'événements et un autre incluant uniquement les événements critiques pour vos instances de production. Vous pouvez facilement désactiver les notifications sans supprimer d'abonnement. Pour ce faire, réglez le bouton radio Activé sur Non dans la console Amazon DocumentDB.

> **Important**
>
> Amazon DocumentDB ne garantie pas l'ordre des événements envoyés dans un flux d'événements. L'ordre des évènements est susceptible de changer.

Amazon DocumentDB utilise l'Amazon Resource Name (ARN) d'une rubrique Amazon SNS pour identifier chaque abonnement. La console Amazon DocumentDB crée l'ARN lorsque vous créez l'abonnement.

La facturation des abonnements aux événements Amazon DocumentDB s'effectue via Amazon SNS. Des frais Amazon SNS s'appliquent en cas d'utilisation de la notification d'évènement. Pour de plus amples informations, veuillez consulter Tarification Amazon Simple Notification Service. Hormis les frais Amazon SNS, Amazon DocumentDB ne facture pas les abonnements aux événements.

Rubriques

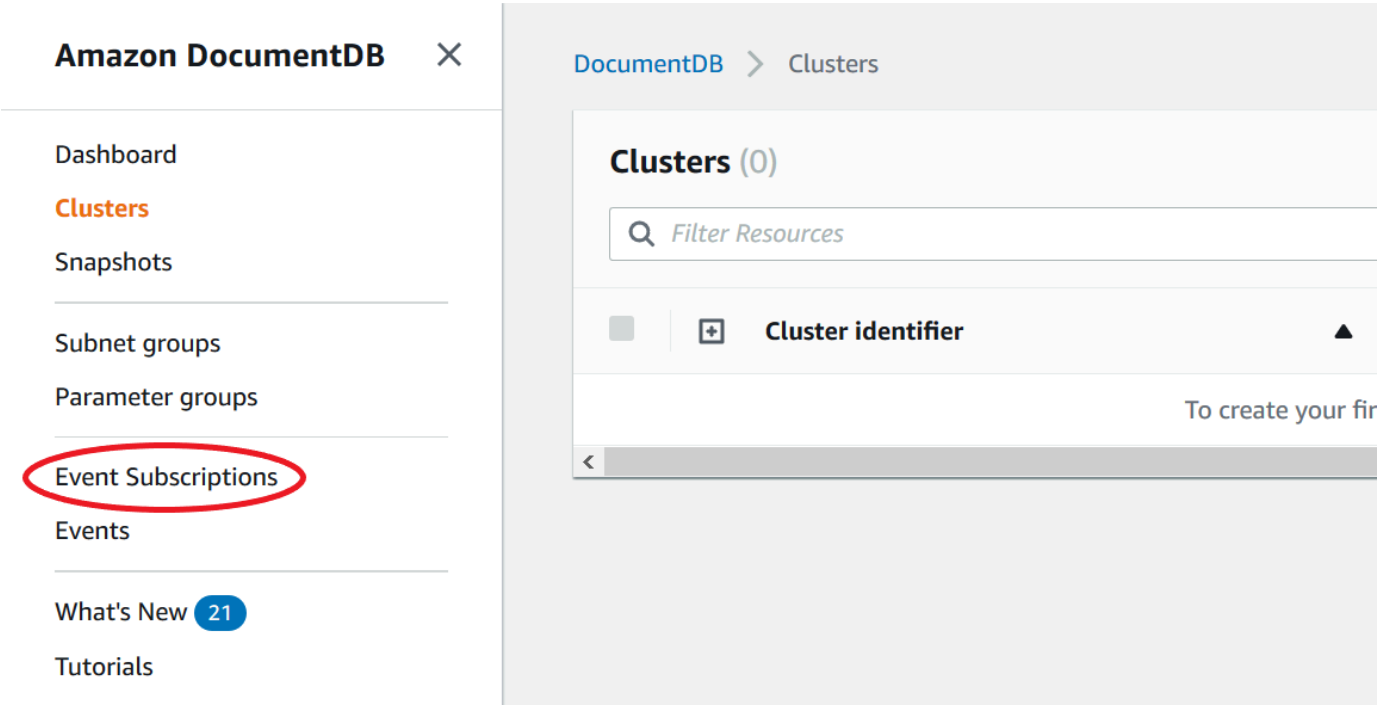
- Abonnement aux abonnements aux événements Amazon DocumentDB
- Gestion des abonnements aux notifications d'événements Amazon DocumentDB
- Catégories d'événements et messages Amazon DocumentDB

Abonnement aux abonnements aux événements Amazon DocumentDB

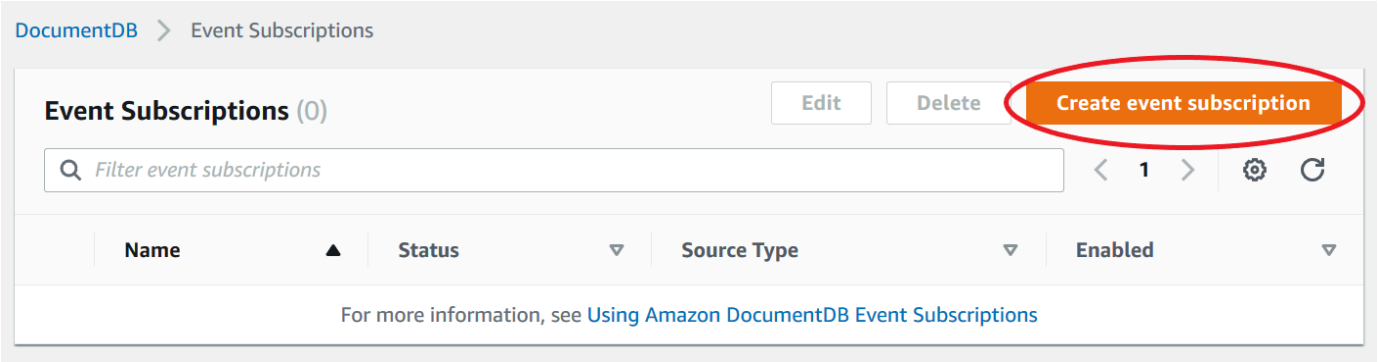
Vous pouvez utiliser la console Amazon DocumentDB pour vous abonner à des événements, comme suit :

1. Connectez-vous à AWS Management Console à l'adresse https://console.aws.amazon.com/docdb.

2. Dans le panneau de navigation, choisissez Abonnements aux évènements.

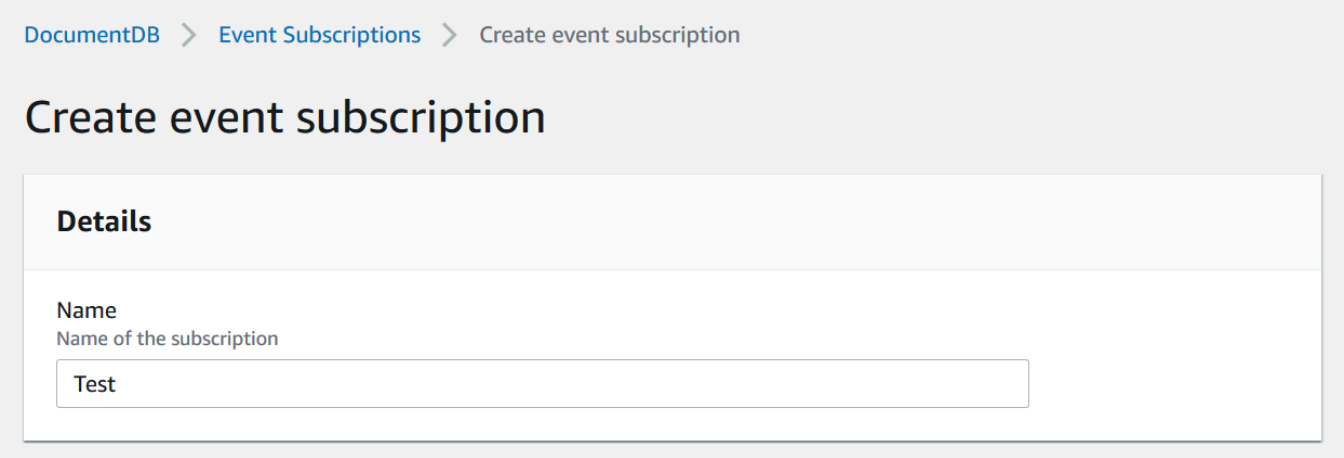


3. Dans le volet Abonnements aux événements, choisissez Créer un abonnement aux événements.

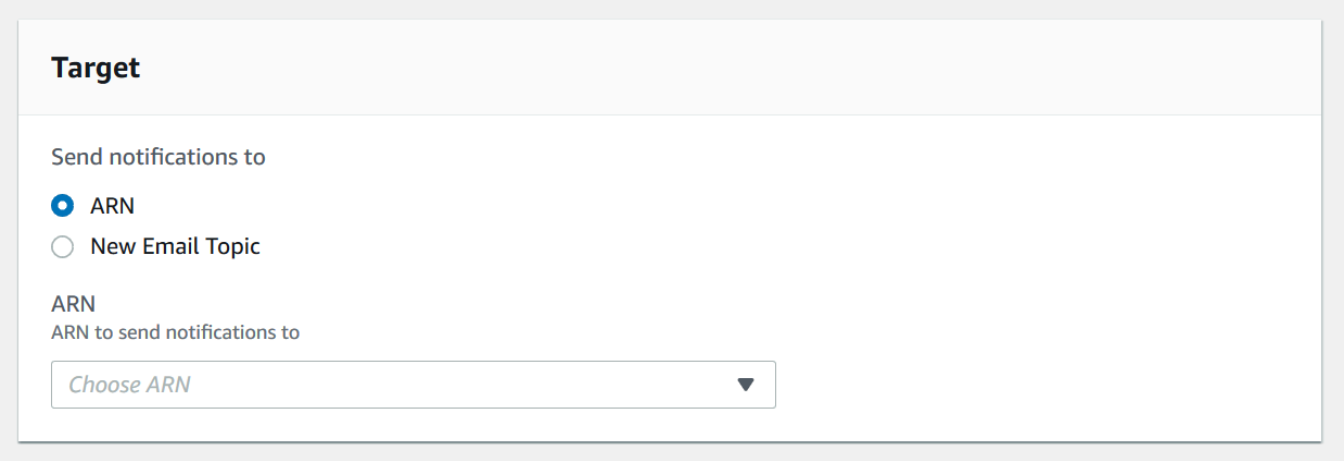


4. Dans la boîte de dialogue Créer un abonnement aux événements, exécutez l'une des actions suivantes :

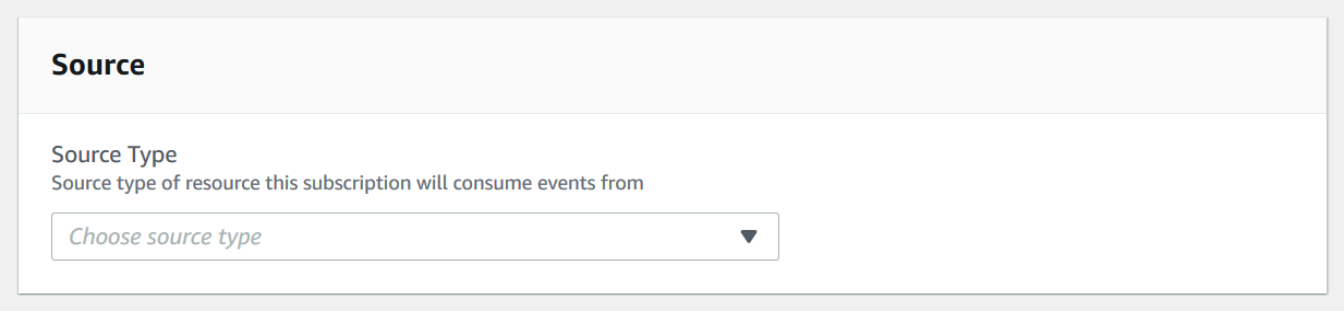
   - Dans le champ Nom, entrez un nom pour l'abonnement à la notification d'événements.



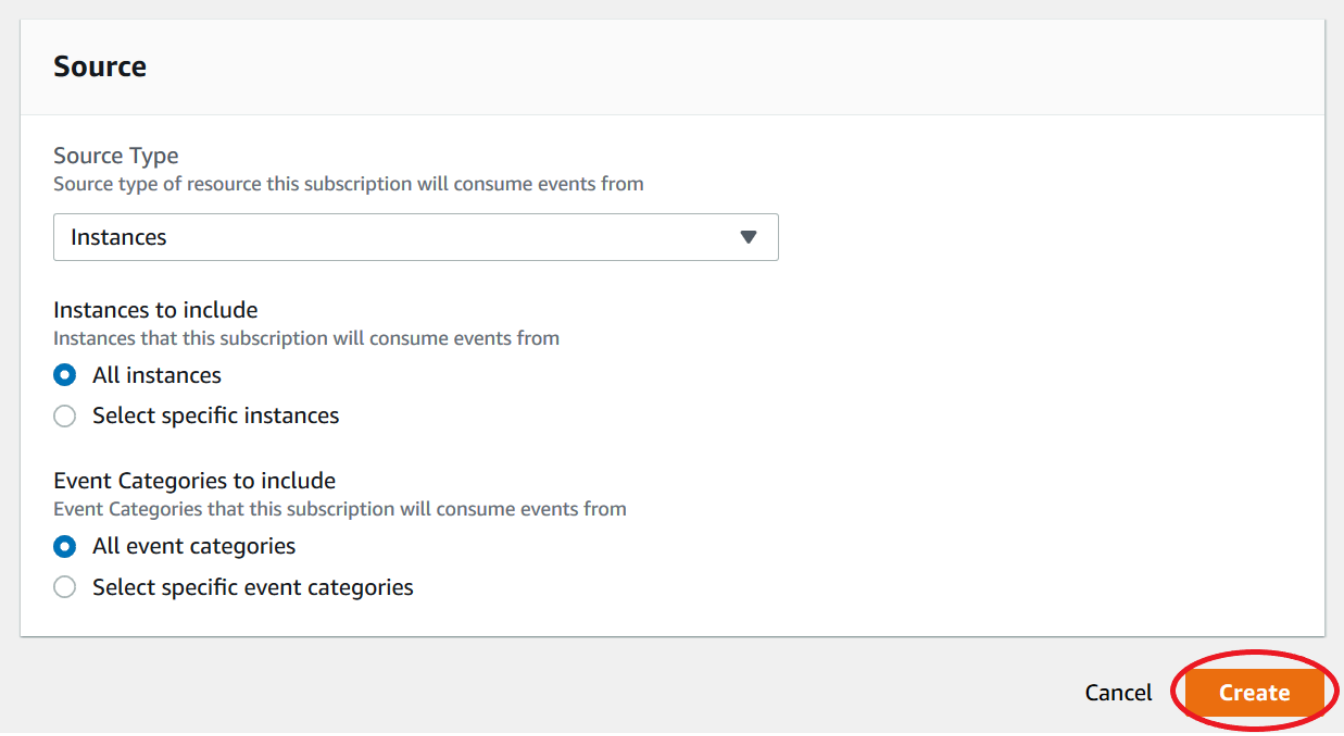
- Pour Target, choisissez l'adresse à laquelle vous souhaitez envoyer les notifications. Vous pouvez choisir un ARN existant ou choisir Nouvelle rubrique d'e-mail pour entrer le nom d'une rubrique et une liste de destinataires.



- Pour Source, choisissez un type de source. Selon le type de source que vous avez sélectionné, choisissez les catégories d'événements et les sources dont vous souhaitez recevoir des notifications d'événements.



- Sélectionnez Create (Créer).

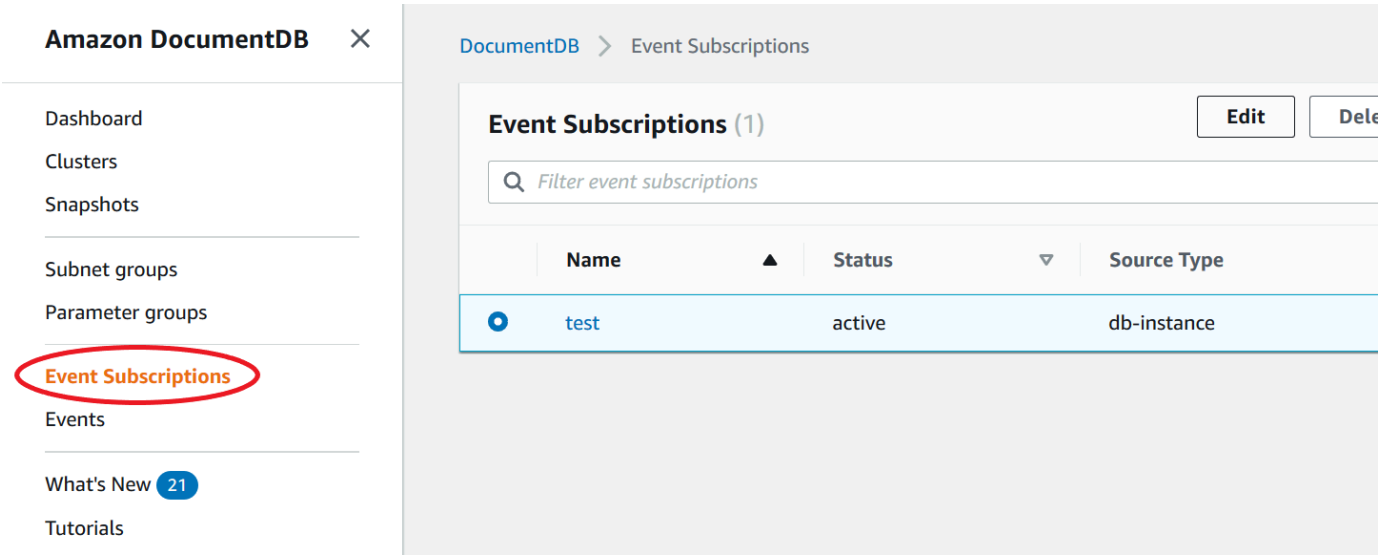


Gestion des abonnements aux notifications d'événements Amazon DocumentDB

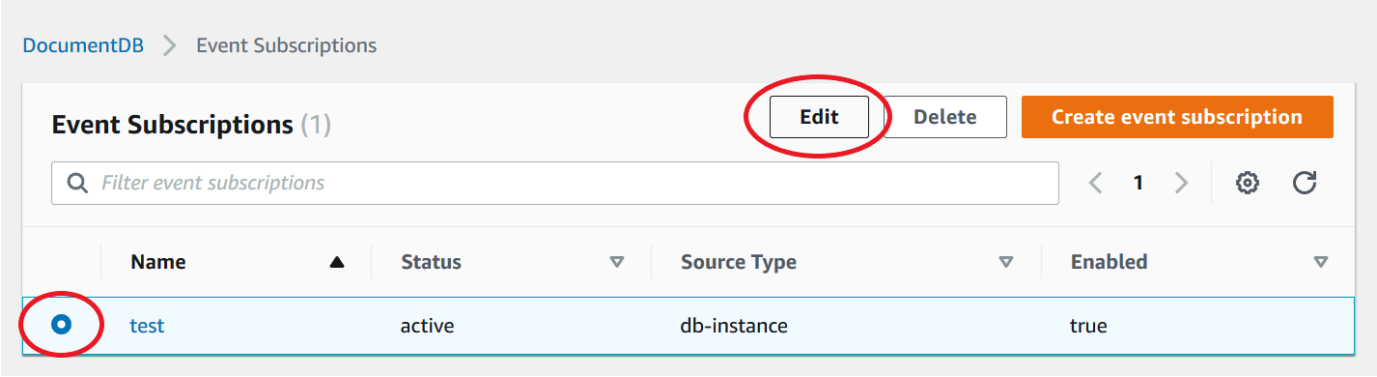
Si vous choisissez Abonnements aux événements dans le volet de navigation de la console Amazon DocumentDB, vous pouvez consulter les catégories d'abonnements et la liste de vos abonnements actuels. Vous pouvez également modifier ou supprimer un abonnement spécifique.

Pour modifier vos abonnements aux notifications d'événements Amazon DocumentDB

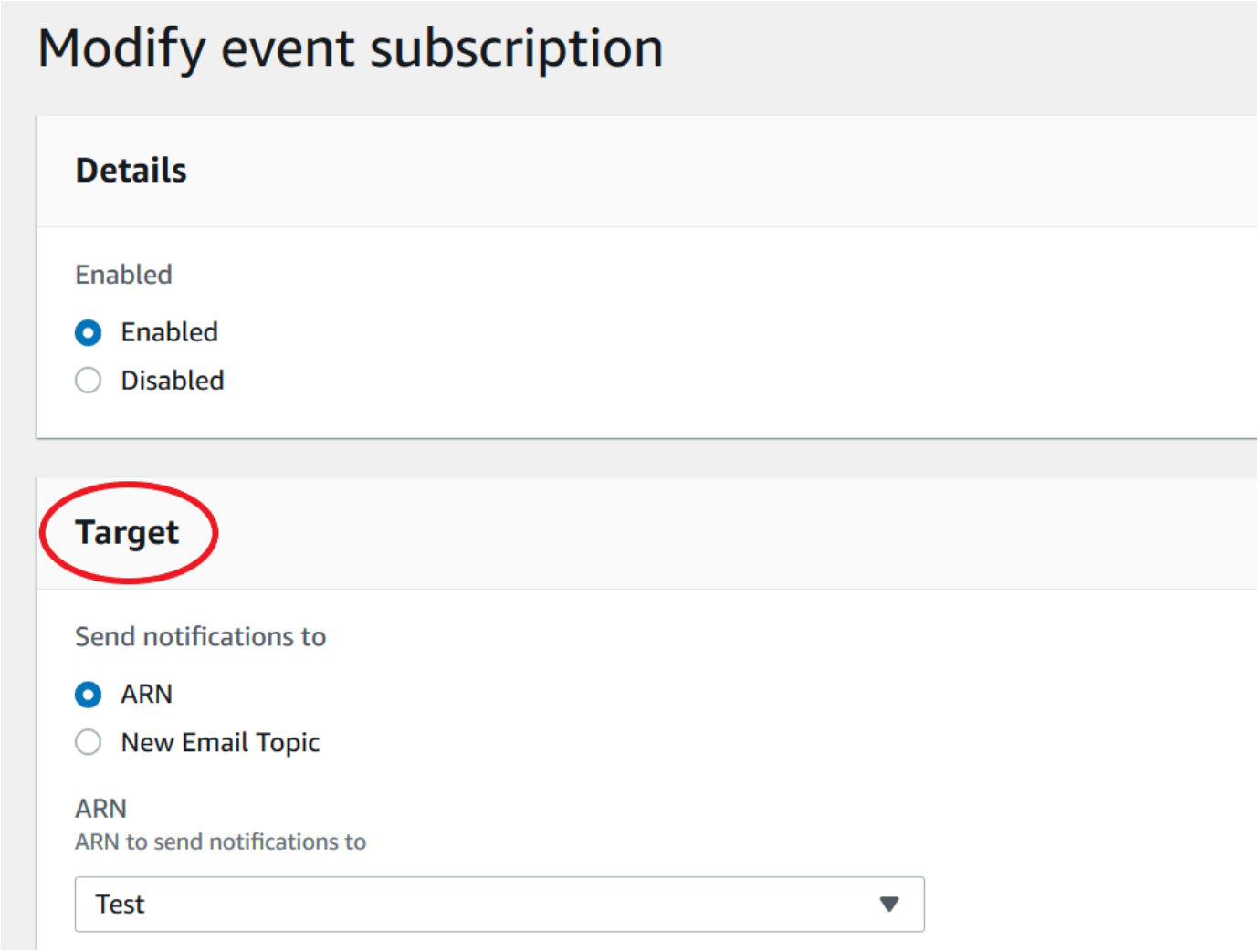
1. Connectez-vous à AWS Management Console à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Abonnements aux évènements. Le volet Abonnements aux événements affiche tous les abonnements aux notifications d'événements.



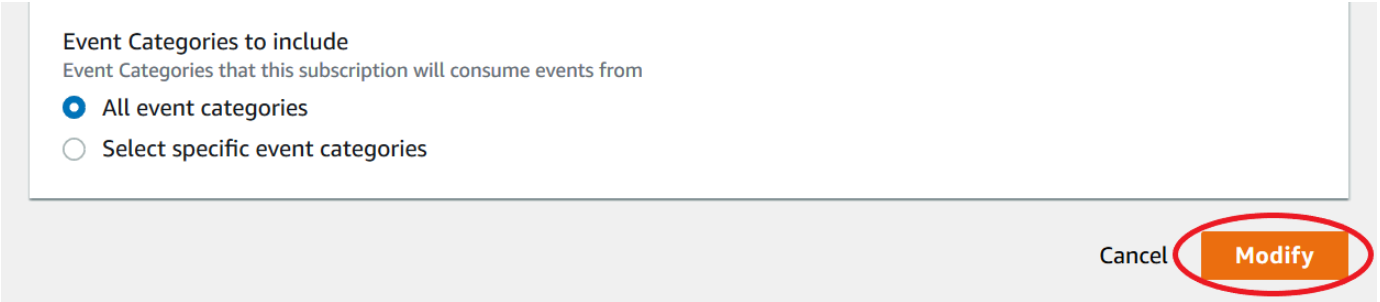
3. Dans le volet Abonnements aux événements, choisissez l'abonnement que vous voulez modifier, puis choisissez Modifier.



4. Apportez les modifications requises à l'abonnement dans la section Cible ou Source. Vous pouvez ajouter ou supprimer des identificateurs source en les sélectionnant ou en annulant leur sélection dans la section Source.



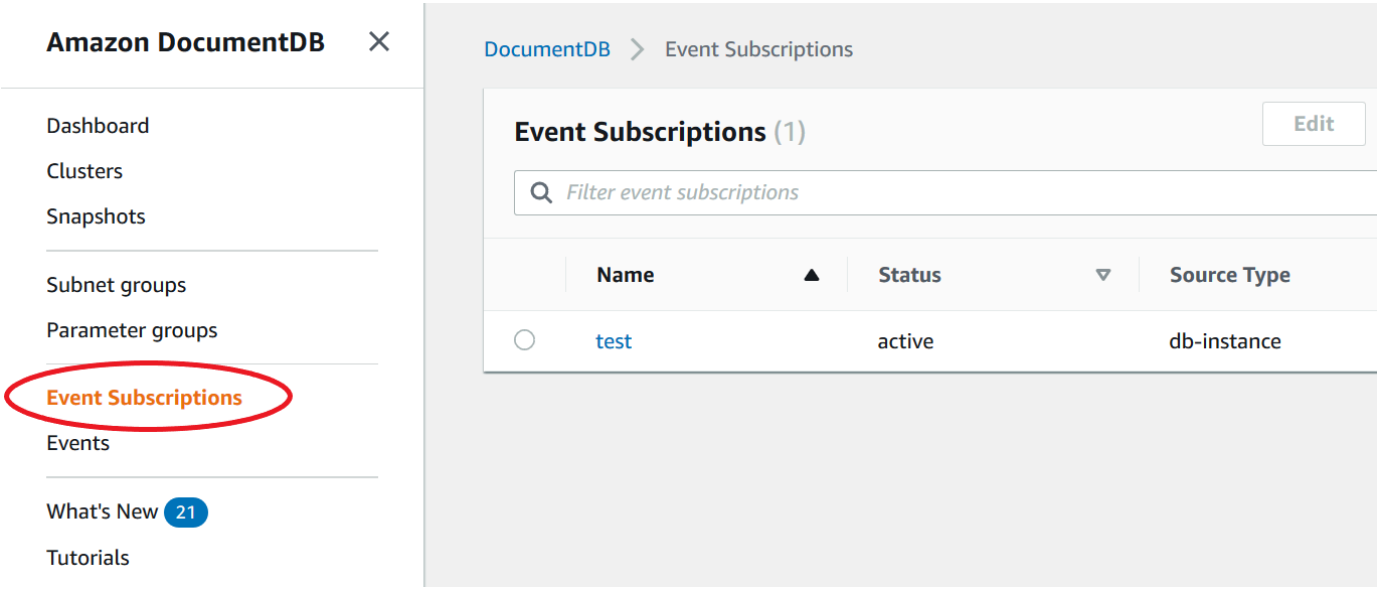
5. Sélectionnez Modifier. La console Amazon DocumentDB indique que l'abonnement est en cours de modification.



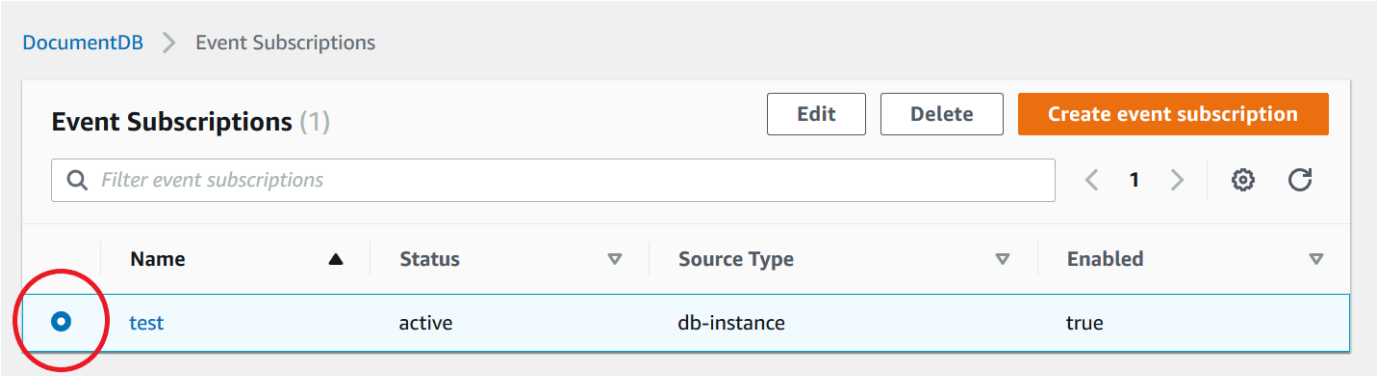
Suppression d'un abonnement aux notifications d'événements Amazon DocumentDB

Vous pouvez supprimer un abonnement lorsque vous n'en avez plus besoin. Tous les abonnés à la rubrique ne reçoivent plus les notifications d'évènements spécifiées par l'abonnement.

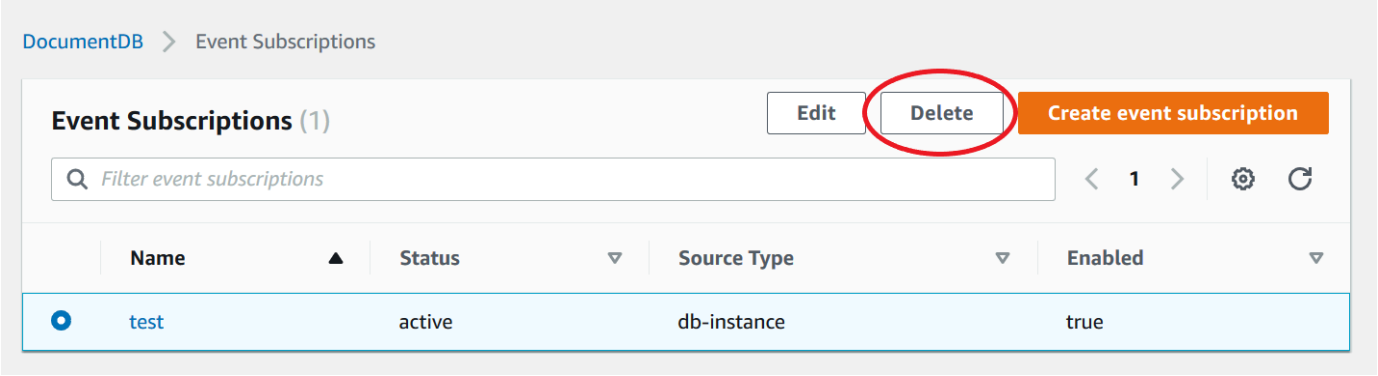
1. Connectez-vous à AWS Management Console à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Abonnements aux évènements.



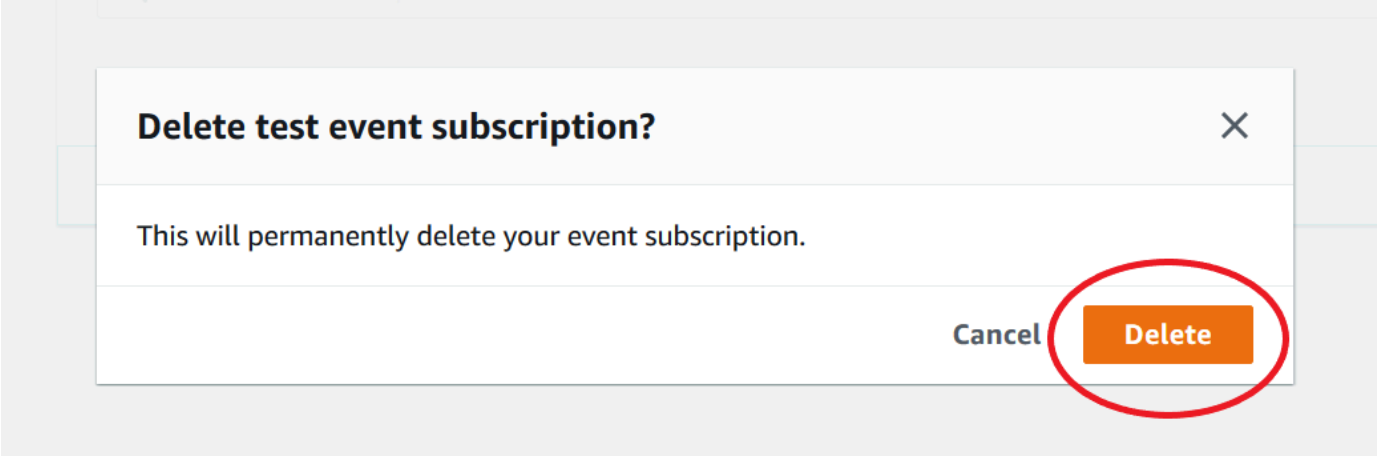
3. Dans le volet Abonnements aux événements, choisissez l'abonnement que vous voulez supprimer.



4. Choisissez Delete (Supprimer).



5. Une fenêtre contextuelle apparaît vous demandant si vous souhaitez supprimer définitivement cette notification. Choisissez Delete (Supprimer).



Catégories d'événements et messages Amazon DocumentDB

Amazon DocumentDB génère un nombre significatif d'événements dans les catégories auxquelles vous pouvez abonner à l'aide de la console. Chaque catégorie s'applique à un type source, qui peut être une instance, un instantané ou un groupe de base de données.

> **Note**
>
> Amazon DocumentDB utilise les définitions et les identifiants d'événements Amazon RDS existants.

Événements Amazon DocumentDB provenant d'instances

Catégorie	Description
disponibilité	L'instance a redémarré.
disponibilité	L'instance s'est arrêtée.
modification de configuration	Application de la modification à une classe d'instance.

Catégorie	Description
modification de configuration	Fin de l'application de la modification à une classe d'instance.
modification de configuration	Réinitialisez les informations d'identification principales.
création	Instance créée.
suppression	Instance supprimée
échec	L'instance a échoué en raison d'une configura tion incompatible ou d'un problème de stockage sous-jacent. Commencez par point-in-time-rest ore pour l'instance.
notification	Instance s'est arrêtée.
notification	Instance démarrée.
notification	L'instance est en cours de démarrage dans la mesure où elle a dépassé le temps maximum autorisé pour son arrêt.
récupération	La récupération de l'instance a commencé. Le temps de récupération varie selon la quantité de données à restaurer.
récupération	La récupération de l'instance est terminée.
application de correctifs de sécurité	La mise à jour du système d'exploitation est disponible pour votre instance. Pour obtenir des informations sur l'application de mises à jour, consultez Gestion d'Amazon DocumentDB.

Événements Amazon DocumentDB provenant d'un cluster

Catégorie	Description
création	cluster créé
suppression	cluster supprimé.
basculement	Promouvoir à nouveau la primaire précédente.
basculement	Basculement vers l'instance terminé.
basculement	Démarrage du basculement vers l'instance de base de données : %s
basculement	Le basculement en mode AZ vers l'instance de base de données a démarré : %s
basculement	Le basculement entre AZ et l'instance de base de données a commencé : %s
maintenance	Le cluster a été corrigé.
maintenance	Le cluster de base de données est dans un état qui ne peut pas être mis à niveau : %s
notification	Le cluster s'est arrêté.
notification	Le cluster a démarré.
notification	L'arrêt du cluster a échoué.
notification	Le cluster est en cours de démarrage dans la mesure où il a dépassé le temps maximum autorisé pour son arrêt.
notification	Cluster renommé de %s à %s.

Événements Amazon DocumentDB provenant d'un instantané du cluster

Le tableau suivant affiche la catégorie d'événement et la liste des événements lorsqu'un instantané de cluster Amazon DocumentDB est le type source.

Catégorie	Description
sauvegarde	Création d'instantané de cluster manuel.
sauvegarde	Instantané de cluster manuel créé.
sauvegarde	Création d'instantané de cluster automatisé.
sauvegarde	Instantané de cluster automatisé créé.

Événements Amazon DocumentDB provenant d'un groupe de paramètres

Le tableau suivant affiche la catégorie d'événement et la liste d'événements quand un groupe de paramètres est le type source.

Catégorie	Description
modification de configuration	Paramètre %s mis à jour vers %s avec la méthode d'application %s

Surveillance d'Amazon DocumentDB avec CloudWatch

Amazon DocumentDB (compatible avec MongoDB) s'intègre à Amazon CloudWatch afin que vous puissiez recueillir et analyser les indicateurs opérationnels de vos clusters. Vous pouvez surveiller ces indicateurs à l'aide du CloudWatch console, la console Amazon DocumentDB, leAWS Command Line Interface(AWS CLI), ou le CloudWatchAPI.

CloudWatch vous permet également de définir des alarmes afin d'être averti si une valeur métrique dépasse un seuil que vous spécifiez. Vous pouvez même configurer Amazon CloudWatch Événements permettant de prendre des mesures correctives en cas de violation. Pour plus d'informations sur l'utilisation CloudWatch et alarmes, consultez leAmazon CloudWatch documentation.

Rubriques

Métriques Amazon DocumentDB

Pour surveiller l'état et les performances de votre cluster et de vos instances Amazon DocumentDB, vous pouvez consulter les métriques suivantes dans la console Amazon DocumentDB.

> **Note**
>
> Les mesures présentées dans les tableaux suivants s'appliquent à la fois aux clusters basés sur des instances et aux clusters élastiques.

Utilisation des ressources

Métrique	Description	
`BackupRetentionPeriodStorageUsed`	La quantité totale de stockage de sauvegarde en Go utilisée pour prendre en charge point-in-time fonctionnalité de restauration dans la fenêtre de rétention d'Amazon DocumentDB. Incluse dans le total indiqué par la métrique `TotalBackupStorageBilled` . Calculée séparément pour chaque cluster Amazon DocumentDB.	

Métrique	Description
`ChangeStreamLogSize`	Quantité de stockage (en Mo) utilisée par votre cluster pour stocker le journal du flux de modifications. Cette valeur est un sous-ensemble du stockage total du cluster (`VolumeBytesUsed`) et affecte le coût du cluster. Pour obtenir des informations sur les tarifs de stockage, consultez lePage produit Amazon DocumentDB. La taille du journal du flux de modifications est fonction de la quantité de modifications qui se produisent sur votre cluster et de la longue durée de rétention du flux de modifications. Pour de plus amples informations sur les flux de modifications, veuillez consulter Utilisation de Change Streams avec Amazon DocumentDB.
`CPUUtilization`	Pourcentage de l'UC utilisé par une instance.
`DatabaseConnections`	Le nombre de connexions ouvertes sur une instance prise à une fréquence d'une minute.

Métrique	Description
DatabaseConnectionsMax	Nombre maximal de connexions de base de données ouvertes sur une instance au cours d'une période d'une minute.
DatabaseCursors	Le nombre de curseurs ouverts sur une instance prise à une fréquence d'une minute.
DatabaseCursorsMax	Nombre maximal de curseurs ouverts sur une instance sur une période d'une minute.
DatabaseCursorsTimedOut	Le nombre de curseurs dont le délai a expiré sur une période d'une minute.
FreeableMemory	Quantité de mémoire vive disponible, en octets.
FreeLocalStorage	Cette métrique indique la quantité de stockage accessible à chaque instance pour les journaux et les tables temporaires. Cette valeur dépend de la classe d'instance. Vous pouvez augmenter la quantité d'espace de stockage libre pour une instance en choisissant une classe d'instance plus grande pour votre instance.

Métrique	Description
`LowMemThrottleQueueDepth`	La profondeur de la file d'attente pour les demandes limitées en raison d'une faible quantité de mémoire disponible traitées à une fréquence d'une minute.
`LowMemThrottleMaxQueueDepth`	Profondeur de file d'attente maximale pour les demandes limitées en raison d'un manque de mémoire disponible sur une période d'une minute.
`LowMemNumOperationsThrottled`	Nombre de demandes limitées en raison d'un manque de mémoire disponible sur une période d'une minute.
`SnapshotStorageUsed`	Quantité totale de stockage de sauvegarde en Go consommée par tous les instantanés d'un cluster Amazon DocumentDB donné en dehors de sa fenêtre de conservation des sauvegardes. Incluse dans le total indiqué par la métrique `TotalBackupStorageBilled` . Calculée séparément pour chaque cluster Amazon DocumentDB.

Métrique	Description
SwapUsage	Quantité d'espace d'échange utilisé sur l'instance.
TotalBackupStorageBilled	La quantité totale de stockage de sauvegarde en GiB pour laquelle vous êtes facturé pour un cluster Amazon DocumentDB donné. Inclut le stockage de sauvegarde mesuré par les métriques BackupRetentionPeriodStorageUsed et SnapshotStorageUsed . Calculée séparément pour chaque cluster Amazon DocumentDB.
TransactionsOpen	Le nombre de transactions ouvertes sur une instance prise à une fréquence d'une minute.
TransactionsOpenMax	Le nombre maximum de transactions ouvertes sur une instance au cours d'une période d'une minute.
VolumeBytesUsed	Volume de stockage, en octets, utilisé par votre cluster. Cette valeur a une incidence sur le coût du cluster. Pour obtenir des informations sur les prix, consultez lePage produit Amazon DocumentDB.

Latence

Métrique	Description
`DBClusterReplicaLagMaximum`	Le délai maximal, en millisecondes, entre l'instance principale et chaque instance Amazon DocumentDB du cluster.
`DBClusterReplicaLagMinimum`	Durée minimale du retard, millisecondes, entre l'instance principale et chaque instance de réplica dans le cluster.
`DBInstanceReplicaLag`	La durée du retard, en millisecondes, lors de la réplication des mises à jour à partir de l'instance principale vers une instance de réplica.
`ReadLatency`	Temps moyen nécessaire pour les opérations d'I/O par disque.
`WriteLatency`	Temps moyen, en millisecondes, nécessaire pour les opérations d'E/S par disque.

Opérations

Métrique	Description
`DocumentsDeleted`	Le nombre de documents supprimés sur une période d'une minute.

Métrique	Description
DocumentsInserted	Le nombre de documents insérés au cours d'une période d'une minute.
DocumentsReturned	Le nombre de documents renvoyés sur une période d'une minute.
DocumentsUpdated	Le nombre de documents mis à jour sur une période d'une minute.
OpcountersCommand	Le nombre de commandes émises au cours d'une période d'une minute.
OpcountersDelete	Nombre d'opérations de suppression effectuées au cours d'une période d'une minute.
OpcountersGetmore	Le nombre de getmores émis sur une période d'une minute.
OpcountersInsert	Nombre d'opérations d'insertion effectuées au cours d'une période d'une minute.
OpcountersQuery	Le nombre de requêtes émises au cours d'une période d'une minute.
OpcountersUpdate	Nombre d'opérations de mise à jour effectuées au cours d'une période d'une minute.

Métrique	Description
`TransactionsStarted`	Le nombre de transactions démarrées sur une instance au cours d'une période d'une minute.
`TransactionsCommitted`	Le nombre de transactions validées sur une instance au cours d'une période d'une minute.
`TransactionsAborted`	Le nombre de transactions abandonnées sur une instance au cours d'une période d'une minute.
`TTLDeletedDocuments`	Le nombre de documents supprimés par un moniteur TTLMonitor sur une période d'une minute.

Débit

Métrique	Description
`NetworkReceiveThroughput`	Quantité de débit réseau reçue des clients par chaque instance du cluster de base de données, en octets par seconde. Ce débit n'inclut pas le trafic réseau entre les instances du cluster et le volume de cluster.
`NetworkThroughput`	Le débit réseau, en octets par seconde, reçu et transmis aux clients par chaque instance du

Métrique	Description
	cluster Amazon DocumentDB. Ce débit n'inclut pas le trafic réseau entre les instances du cluster et le volume de cluster.
`NetworkTransmitThroughput`	Quantité de débit réseau envoyée aux clients par chaque instance du cluster, en octets par seconde. Ce débit n'inclut pas le trafic réseau entre les instances du cluster et le volume de cluster.
`ReadIOPS`	Nombre moyen d'opérations d'I/O de lecture de disque par seconde. Amazon DocumentDB rapporte les IOPS en lecture et en écriture séparément, à intervalles d'une minute.
`ReadThroughput`	Nombre moyen d'octets lus sur le disque par seconde.

Métrique	Description	
VolumeReadIOPs	Nombre moyen d'opérations d'E/S de lecture facturées depuis un volume de cluster, rapportées par intervalles de 5 minutes. Les opérations lues facturées sont calculées au niveau du volume de cluster, regroupées à partir de toutes les instances du cluster, puis rapportées par intervalles de 5 minutes. La valeur est calculée en prenant la valeur de la métrique des opérations de lecture sur une période de 5 minutes. Vous pouvez déterminer la quantité d'opérations lues facturées par seconde en prenant la valeur de la métrique des opération s de lecture facturées et en la divisant par 300 secondes. Par exemple, si leVolumeRea dIOPs renvoie 13 686, puis le nombre d'opérations de lecture facturées par seconde est de 45 (13 686/300 = 45,62). Vous cumulez les opérations de lecture facturées pour les requêtes qui demandent des pages de base de données non présentes dans le cache des tampons et qui doivent,	

Métrique	Description	
	par conséquent, être chargées depuis le stockage. Il se peut que vous constatiez des pics dans les opérations de lecture facturées, car les résultats des requêtes sont lus à partir du stockage, puis chargés dans le cache des tampons.	

Métrique	Description
`VolumeWriteIOPs`	Nombre moyen d'opérations d'E/S de lecture facturées depuis un volume de cluster, rapportées par intervalles de 5 minutes. Les opérations lues facturées sont calculées au niveau du volume de cluster, regroupées à partir de toutes les instances du cluster, puis rapportées par intervalles de 5 minutes. La valeur est calculée en prenant la valeur de la métrique des opérations en écriture sur une période de 5 minutes. Vous pouvez déterminer la quantité d'opérations lues facturées par seconde en prenant la valeur de la métrique des opération s en écriture facturées et en la divisant par 300 secondes. Par exemple, si le`VolumeWri teIOPs` renvoie 13 686, puis le nombre d'opérati ons d'écriture facturées par seconde est de 45 (13 686/300 = 45,62). Notez que`VolumeRea dIOPs` et`VolumeWri teIOPs` les métriques sont calculées par la couche de stockage DocumentDB et incluent iOS exécuté par

Métrique	Description	
	les instances principales et répliquées. Les données sont agrégées toutes les 20 à 30 minutes, puis rapportées à intervalles de 5 minutes, émettant ainsi le même point de données pour la métrique au cours de la période. Si vous recherchez une métrique à corréler à vos opérations d'insertion sur un intervalle d'une minute, vous pouvez utiliser la métrique WriteIops au niveau de l'instance. La métrique est disponible dans l'onglet de surveillance de votre instance principale Amazon DocumentDB.	
`WriteIOPS`	Nombre moyen d'opérations d'I/O d'écriture de disque par seconde. Lorsqu'il est utilisé au niveau du cluster,`WriteIOPs` sont évalués sur toutes les instances du cluster. Les IOPS de lecture et d'écriture sont signalées séparément et à des intervalles d'une minute.	
`WriteThroughput`	Nombre moyen d'octets écrits sur le disque par seconde.	

Système

Métrique	Description
`BufferCacheHitRatio`	Pourcentage de demandes traitées par le cache de tampons.
`DiskQueueDepth`	le nombre de demandes d'écriture simultanées sur le volume de stockage distribué.
`EngineUptime`	Temps d'exécution de l'instance, en secondes.
`IndexBufferCacheHitRatio`	Pourcentage de demandes d'index traitées par le cache tampon. Il se peut que vous observiez un pic supérieur à 100 % pour la métrique juste après la suppression d'un index, d'une collection ou d'une base de données. Cela sera automatiquement corrigé au bout de 60 secondes. Cette limitation sera corrigée lors d'une prochaine mise à jour du correctif.

Métriques de l'instance T3

Métrique	Description
`CPUCreditUsage`	Le nombre de crédits CPU dépensés pendant la période de mesure.

Métrique	Description
`CPUCreditBalance`	Le nombre de crédits CPU accumulés par une instance. Ce solde diminue lorsque les crédits UC sont dépensés plus rapidement qu'ils ne sont gagnés.
`CPUSurplusCreditBalance`	Le nombre de crédits CPU excédentaires dépensés pour maintenir les performances du processeur lorsque le processeurCreditBalance la valeur est zéro.
`CPUSurplusCreditsCharged`	Le nombre de crédits CPU excédentaires dépassant le nombre maximum de crédits CPU pouvant être gagnés sur une période de 24 heures, et entraînant ainsi des frais supplémentaires. Pour plus d'informations, voirSurveillance des crédits de votre processeur.

Visualisation CloudWatch Données

Vous pouvez consulter Amazon CloudWatch données utilisant le CloudWatch console, la console Amazon DocumentDB,AWS Command Line Interface(AWS CLI), ou le CloudWatch API.

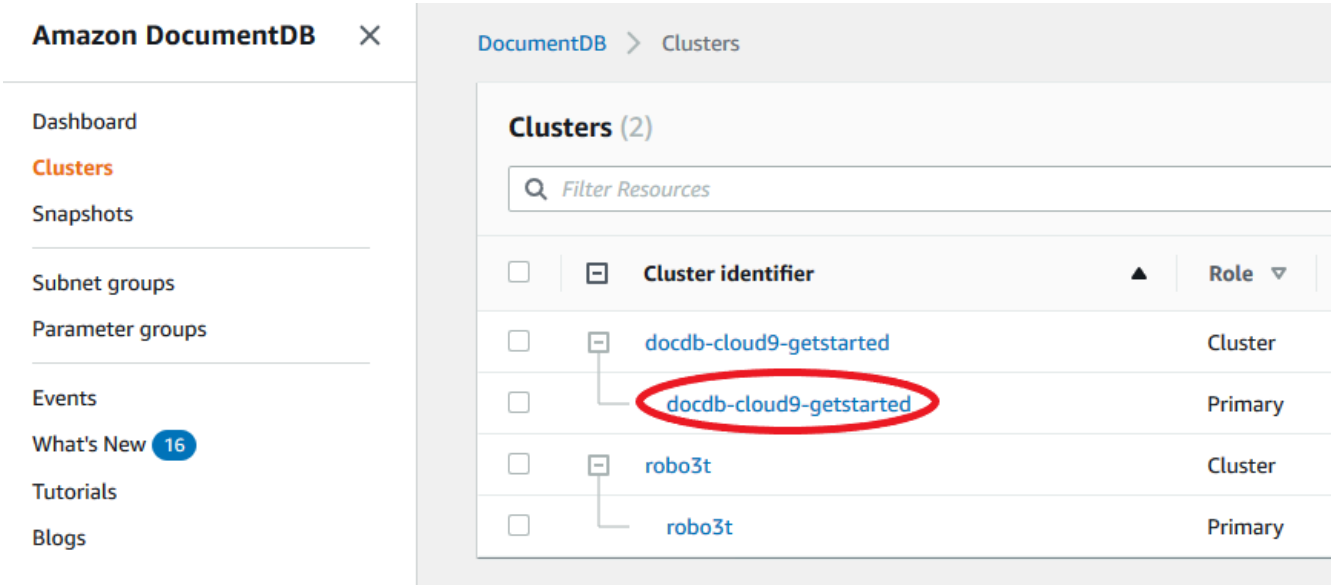
Using the AWS Management Console

Pour voir CloudWatch métriques à l'aide de la console de gestion Amazon DocumentDB, procédez comme suit.

1. Connectez-vous auAWS Management Console, et ouvrez la console Amazon DocumentDB à l'adressehttps://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters.

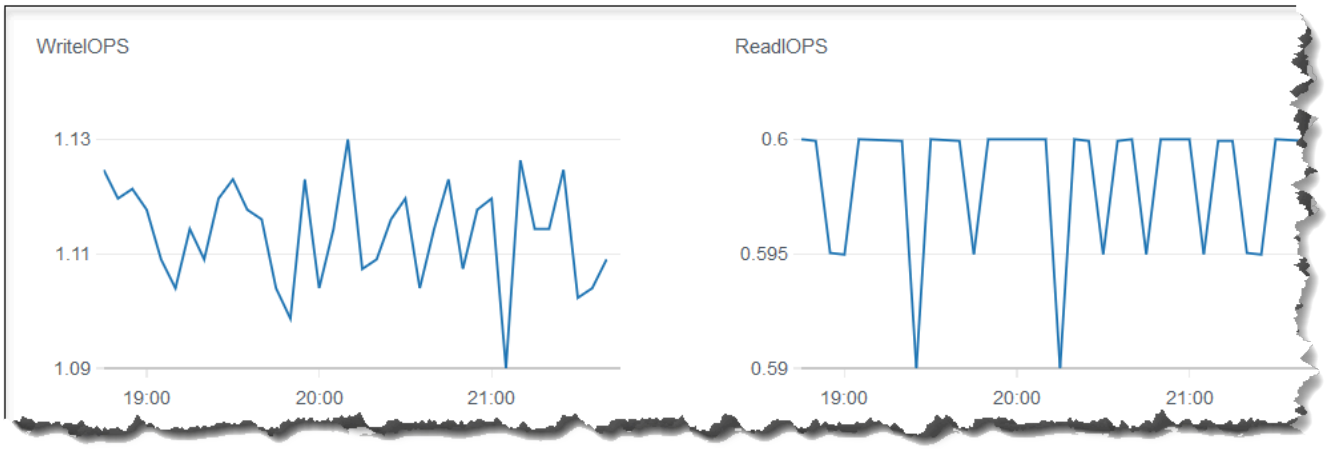
   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans la zone de navigation Clusters, vous verrez la colonneIdentifiant du cluster. Vos instances sont répertoriées sous des clusters, comme dans la capture d'écran ci-dessous.



4. Dans la liste des instances, choisissez le nom de l'instance pour laquelle vous souhaitez obtenir des métriques.
5. Dans la page de résumé de l'instance qui s'affiche, sélectionnezSurveillanceonglet pour afficher les représentations graphiques des métriques de votre instance Amazon DocumentDB. Étant donné qu'un graphique doit être généré pour chaque métrique, leCloudWatchgraphiques à remplir.

   L'image suivante montre les représentations graphiques de deux CloudWatch métriques dans la console Amazon DocumentDB,`WriteIOPS`et`ReadIOPS`.



Using the CloudWatch Management Console

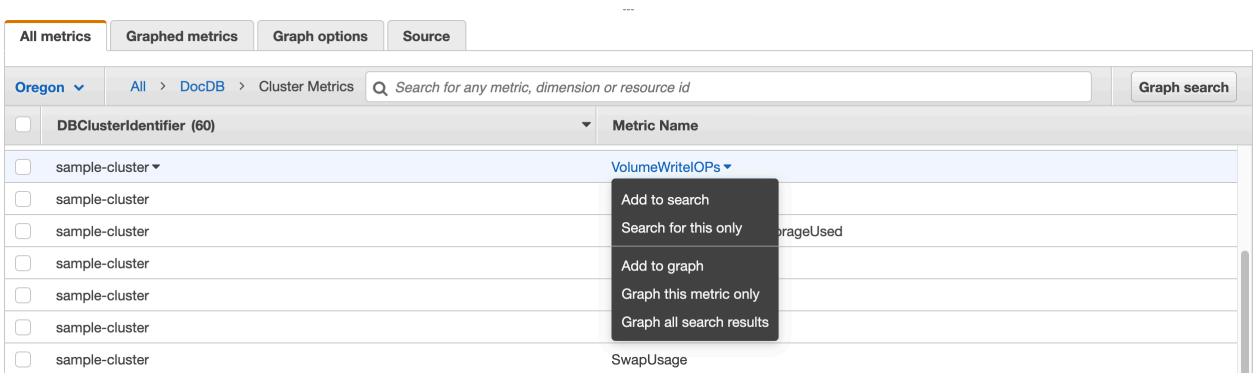
Pour voir CloudWatch métriques utilisant le CloudWatch Console de gestion, effectuez les étapes suivantes.

1. Connectez-vous auAWS Management Console, et ouvrez la console Amazon DocumentDB à l'adressehttps://console.aws.amazon.com/cloudwatch.
2. Dans le panneau de navigation, sélectionnez Metrics (Métriques). Ensuite, dans la liste des noms de service, choisissezDocDB.
3. Choisissez une dimension métrique (par exemple,Métriques du cluster).
4. LeToutes les métriquesl'onglet affiche toutes les mesures pour cette dimension dansDocDB.

    a. Pour trier le tableau, utilisez l'en-tête de colonne.

    b. Pour représenter graphiquement une métrique, cochez la case en regard de la métrique. Pour sélectionner toutes les métriques, cochez la case dans la ligne d'en-tête du tableau.

    c. Pour filtrer par métrique, passez le curseur sur le nom de la métrique et sélectionnez la flèche déroulante à côté du nom de la métrique. Ensuite, choisissezAjouter à la recherche, comme le montre l'image ci-dessous.



Using the AWS CLI

Pour voir CloudWatch données pour Amazon DocumentDB, utilisez CloudWatch `get-metric-statistics`opération avec les paramètres suivants.

Paramètres

- **`--namespace`** — Obligatoire. L'espace de nom du service pour lequel vous voulez les métriques CloudWatch . Pour Amazon DocumentDB, cela doit être`AWS/DocDB`.
- **`--metric-name`** — Obligatoire. Nom de la métrique pour laquelle vous souhaitez obtenir des données.
- **`--start-time`** — Obligatoire. L'horodatage qui détermine le premier point de données à renvoyer.

  La valeur spécifiée est inclusive ; les résultats incluent des points de données avec l'horodatage spécifié. L'horodatage doit être au format ISO 8601 UTC (par exemple, 2016-10-03T23:00:00Z).
- **`--end-time`** — Obligatoire. L'horodatage qui détermine le dernier point de données à renvoyer.

  La valeur spécifiée est inclusive ; les résultats incluent des points de données avec l'horodatage spécifié. L'horodatage doit être au format ISO 8601 UTC (par exemple, 2016-10-03T23:00:00Z).
- **`--period`** — Obligatoire. La granularité, en secondes, des points de données renvoyés. Pour les métriques avec une résolution standard, une période peut se réduire à une seule minute (60 secondes) et doit être un multiple de 60. Pour les métriques haute résolution qui sont collectées à des intervalles de moins d'une minute, la période peut être 1, 5, 10, 30, 60, ou tout multiple de 60.
- **`--dimensions`**— Facultatif. Si la métrique contient plusieurs dimensions, vous devez inclure une valeur pour chaque dimension. CloudWatch traite chaque combinaison unique de

dimensions comme une métrique distincte. Vous ne pouvez pas récupérer les statistiques d'une combinaison de dimensions qui n'a pas été spécifiquement publiée. Vous devez spécifier les mêmes dimensions que celles utilisées lorsque les mesures ont été créées.

- **`--statistics`**— Facultatif. Statistiques de la métrique, autres que des percentiles. Pour les statistiques sur les centiles, utilisez `ExtendedStatistics`. En appelant `GetMetricStatistics`, vous devez spécifier `Statistics` ou `ExtendedStatistics`, mais pas les deux.

  Valeurs autorisées :

  - `SampleCount`
  - `Average`
  - `Sum`
  - `Minimum`
  - `Maximum`
- **`--extended-statistics`**— Facultatif. Les statistiques sur les `percentile`. Spécifiez des valeurs comprises entre p0.0 et p100. En appelant `GetMetricStatistics`, vous devez spécifier `Statistics` ou `ExtendedStatistics`, mais pas les deux.
- **`--unit`**— Facultatif. L'unité pour une métrique donnée. Les métriques peuvent être exprimées en unités multiples. Le fait de ne pas fournir une unité entraîne le renvoi de toutes les unités. Si vous spécifiez uniquement une unité que la métrique ne rapporte pas, les résultats de l'appel sont null.

  Valeurs possibles :

  - `Seconds`
  - `Microseconds`
  - `Milliseconds`
  - `Bytes`
  - `Kilobytes`
  - `Megabytes`
  - `Gigabytes`
  - `Terabytes`
  - `Bits`
  - `Kilobytes`

- Megabits
- Gigabits
- Terabits
- Percent
- Count
- Bytes/Second
- Kilobytes/Second
- Megabytes/Second
- Gigabytes/Second
- Terabytes/Second
- Bits/Second
- Kilobits/Second
- Megabits/Second
- Gigabits/Second
- Terabits/Second
- Count/Second
- None

Example

L'exemple suivant trouve la CPUUtilization maximale pour une période de 2 heures en prélevant un échantillon toutes les 60 secondes.

Pour Linux, macOS ou Unix :

```
aws cloudwatch get-metric-statistics \
       --namespace AWS/DocDB \
       --dimensions \
           Name=DBInstanceIdentifier,Value=docdb-2019-01-09-23-55-38 \
       --metric-name CPUUtilization \
       --start-time 2019-02-11T05:00:00Z \
       --end-time 2019-02-11T07:00:00Z \
       --period 60 \
       --statistics Maximum
```

Pour Windows :

```
aws cloudwatch get-metric-statistics ^
       --namespace AWS/DocDB ^
       --dimensions ^
           Name=DBInstanceIdentifier,Value=docdb-2019-01-09-23-55-38 ^
       --metric-name CPUUtilization ^
       --start-time 2019-02-11T05:00:00Z ^
       --end-time 2019-02-11T07:00:00Z ^
       --period 60 ^
       --statistics Maximum
```

Le résultat de cette opération ressemble à ce qui suit :

```
{
       "Label": "CPUUtilization",
       "Datapoints": [
           {
               "Unit": "Percent",
               "Maximum": 4.49152542374361,
               "Timestamp": "2019-02-11T05:51:00Z"
           },
           {
               "Unit": "Percent",
               "Maximum": 4.25000000000485,
               "Timestamp": "2019-02-11T06:44:00Z"
           },

           ********* some output omitted for brevity *********

           {
               "Unit": "Percent",
               "Maximum": 4.33333333331878,
               "Timestamp": "2019-02-11T06:07:00Z"
           }
       ]
   }
```

Dimensions d'Amazon DocumentDB

Les métriques d'Amazon DocumentDB sont qualifiées par les valeurs du compte ou de l'opération. Vous pouvez utiliser le CloudWatch console pour récupérer les données Amazon DocumentDB filtrées selon l'une des dimensions du tableau suivant.

Dimension	Description
`DBClusterIdentifier`	Filtre les données que vous demandez pour un cluster Amazon DocumentDB spécifique.
`DBClusterIdentifier, Role`	Filtre les données que vous demandez pour un cluster Amazon DocumentDB spécifique, en agrégeant la métrique par rôle d'instanc e (WRITER/READER). Par exemple, vous pouvez regrouper des métriques pour toutes les instances READER qui appartiennent à un cluster.
`DBInstanceIdentifier`	Filtre les données que vous demandez pour une instance de base de données spécifique.

Surveillance des compteurs

Les métriques Opcounter ont une valeur différente de zéro (généralement ~50) pour les clusters inactifs. Cela est dû au fait qu'Amazon DocumentDB effectue des contrôles de santé périodiques, des opérations internes et des tâches de collecte de métriques.

Surveillance des connexions de base de données

Lorsque vous visualisez le nombre de connexions à l'aide des commandes du moteur de base de données telles que`db.runCommand( { serverStatus: 1 })`, vous pouvez voir jusqu'à 10 connexions de plus que ce que vous voyez dans`DatabaseConnections`à travers CloudWatch. Cela se produit parce qu'Amazon DocumentDB effectue des contrôles de santé périodiques et des tâches de collecte de métriques qui ne sont pas prises en compte dans`DatabaseConnections`.`DatabaseConnections`représente uniquement les connexions initiées par le client.

Journalisation des appels d'API Amazon DocumentDB à l'aide d'AWS CloudTrail

Amazon DocumentDB (avec la compatibilité MongoDB) est intégré avecAWS CloudTrail, service qui enregistre les actions effectuées par les utilisateurs, les rôles ou unAWS service dans Amazon DocumentDB (avec la compatibilité MongoDB). CloudTrail capture tous les appels d'AWS CLIAPI pour Amazon DocumentDB en tant qu'événements, y compris les appels de la console Amazon DocumentDB et les appels de code à Amazon DocumentDB SDK Amazon DocumentDB. Si vous créez un journal d'activité, vous pouvez activer la livraison continue des CloudTrail événements dans un compartiment Amazon S3, y compris les événements pour Amazon DocumentDB. Si vous ne configurez pas de journal d'activité, vous pouvez toujours afficher les événements les plus récents sur la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été envoyée à Amazon DocumentDB (avec la compatibilité MongoDB), l'adresse IP source à partir de laquelle la demande a été effectuée, l'auteur de la demande et la date de la demande, ainsi que d'autres détails.

> **Important**
>
> Pour certaines fonctionnalités de gestion, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon Relational Database Service (Amazon RDS). La console Amazon DocumentDB etAWS CLI les appels d'API sont enregistrés en tant qu'appels effectués vers l'API Amazon RDS.

Pour plus d'informations sur l'utilisation d'AWS CloudTrail, consultez le Guide de l'utilisateur AWS CloudTrail.

Informations sur Amazon DocumentDB dans CloudTrail

CloudTrail est activé dans votreCompte AWS lors de la création de ce dernier. Lorsqu'une activité se produit dans Amazon DocumentDB (avec la compatibilité MongoDB), elle est enregistrée dans un CloudTrail événement avec d'autres événements deAWS services dans Historique des événements. Vous pouvez afficher, rechercher et télécharger les événements récents dans votre Compte AWS. Pour de plus amples informations, veuillez consulter Affichage des événements avec l'historique des CloudTrail événements.

Pour obtenir un registre continu des événements dans votreCompte AWS, y compris les événements pour Amazon DocumentDB (avec la compatibilité MongoDB), créez un journal d'activité. Un journal

CloudTrail de suivi permet de diffuser des fichiers journaux vers un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions Régions AWS. Le journal d'activité consigne les événements de toutes les Régions dans la partition AWS et livre les fichiers journaux dans le compartiment Simple Storage Service (Amazon S3) de votre choix. En outre, vous pouvez configurer d'autresAWS services pour analyser plus en profondeur les données d'événement collectées dans les CloudTrail journaux et agir sur celles-ci. Pour plus d'informations, consultez les rubriques suivantes dans le AWS CloudTrailGuide de l'utilisateur :

- Présentation de la création d'un journal d'activité
- CloudTrail Services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs Régions
- Réception de fichiers CloudTrail journaux de plusieurs comptes

Chaque événement ou entrée du journal contient des informations sur la personne qui a généré la demande. Les informations relatives à l'identité permettent de déterminer :

- Si la demande a été effectuée avec les informations d'identification utilisateur racine ou .
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la requête a été effectuée par un autre service AWS.

Pour plus d'informations, consultez la section Élément userIdentity CloudTrail.

Profilage des opérations Amazon DocumentDB

Vous pouvez utiliser le profileur dans Amazon DocumentDB (compatible avec MongoDB) pour enregistrer le temps d'exécution et les détails des opérations effectuées sur votre cluster. Le profileur est utile pour surveiller les opérations les plus lentes sur votre cluster afin de vous aider à améliorer les performances des requêtes individuelles et les performances globales du cluster.

Par défaut, la fonction de profileur est désactivée. Lorsqu'il est activé, le profileur enregistre les opérations qui prennent plus de temps qu'une valeur de seuil définie par le client (par exemple, 100 ms) sur Amazon CloudWatch Journaux. Les détails consignés incluent la commande profilée,

l'heure, le récapitulatif du plan et les métadonnées du client. Une fois les opérations enregistrées dans CloudWatch Des journaux, que vous pouvez utiliser CloudWatch Logs Insights pour analyser, surveiller et archiver vos données de profilage Amazon DocumentDB. Les requêtes courantes sont fournies dans la section Requêtes courantes.

Lorsque cette option est activée, le profileur utilise des ressources supplémentaires de votre cluster. Nous vous recommandons de commencer avec une valeur de seuil élevée (par exemple, 500 ms) et de réduire progressivement la valeur pour identifier les opérations lentes. Démarrer avec une valeur de seuil de 50 ms peut entraîner des problèmes de performances sur votre cluster pour les applications à haut débit. Le profileur est activé au niveau du cluster et fonctionne sur toutes les instances et bases de données d'un cluster. Amazon DocumentDB enregistre les opérations sur Amazon CloudWatch Se connecte dans la mesure du possible.

Bien qu'Amazon DocumentDB n'impose aucun frais supplémentaire pour activer le profileur, les tarifs standard vous sont facturés pour l'utilisation de CloudWatch Journaux. Pour plus d'informations sur CloudWatch Tarification des journaux, voirAmazon CloudWatch tarification.

Rubriques

Opérations prises en charge

Le profileur Amazon DocumentDB prend en charge les opérations suivantes :

- `aggregate`
- `count`
- `delete`
- `distinct`
- `find` (OP_QUERY et commande)

- `findAndModify`
- `insert`
- `update`

Limites

Le profileur de requêtes lent ne peut émettre des journaux de profilage que si l'ensemble des résultats de la requête peut tenir dans un seul lot et si le jeu de résultats est inférieur à 16 Mo (taille BSON maximale). Les ensembles de résultats supérieurs à 16 Mo sont automatiquement divisés en plusieurs lots.

La plupart des pilotes ou des interpréteurs de commandes peuvent définir une taille de lot par défaut faible. Vous pouvez spécifier la taille du lot dans le cadre de votre requête. Afin de capturer des journaux de requêtes lents, nous recommandons une taille de lot supérieure à la taille de votre jeu de résultats attendu. Si vous n'êtes pas sûr de la taille du jeu de résultats ou si elle varie, vous pouvez également définir la taille du lot sur un grand nombre (par exemple, 100 000).

Cependant, l'utilisation d'une taille de lot plus importante signifie que davantage de résultats devront être extraits de la base de données avant qu'une réponse ne soit envoyée au client. Pour certaines requêtes, cela peut entraîner des délais plus longs avant que vous n'obteniez des résultats. Si vous ne prévoyez pas de consommer l'ensemble des résultats, il est possible que vous dépensiez davantage d'E/S pour traiter la requête et rejeter le résultat.

Activation du profileur Amazon DocumentDB

L'activation du profileur sur un cluster est un processus en trois étapes. Assurez-vous que toutes les étapes sont terminées, sinon les journaux de profilage ne seront pas envoyés à CloudWatch Journaux. Le profileur est défini au niveau du cluster et est effectué sur l'ensemble des bases de données et des instances du cluster.

Pour activer le profileur sur un cluster

1. Etant donné que vous ne pouvez pas modifier un groupe de paramètres de cluster par défaut, veillez à disposer d'un groupe de paramètres de cluster personnalisé disponible. Pour plus d'informations, veuillez consulter Création de groupes de paramètres de cluster Amazon DocumentDB.
2. À l'aide d'un groupe de paramètres de cluster personnalisé disponible, modifiez les paramètres suivants : `profiler`, `profiler_threshold_ms` et `profiler_sampling_rate`. Pour plus

d'informations, veuillez consulter Modification des groupes de paramètres du cluster Amazon DocumentDB.

3. Créez ou modifiez votre cluster pour utiliser le groupe de paramètres de cluster personnalisé et pour activer l'exportation`profiler`se connecte à CloudWatch Journaux.

Les sections suivantes montrent comment implémenter ces étapes à l'aide d'AWS Management Console et de l'AWS Command Line Interface (AWS CLI).

Using the AWS Management Console

1. Avant de commencer, créez un cluster Amazon DocumentDB et un groupe de paramètres de cluster personnalisé si vous n'en avez pas déjà un. Pour plus d'informations, consultez Création de groupes de paramètres de cluster Amazon DocumentDB et Création d'un cluster Amazon DocumentDB.
2. À l'aide d'un groupe de paramètres de cluster personnalisé disponible, modifiez les paramètres suivants. Pour plus d'informations, veuillez consulter Modification des groupes de paramètres du cluster Amazon DocumentDB.

   - `profiler`— Active ou désactive le profilage des requêtes. Les valeurs autorisées sont `enabled` et `disabled`. La valeur par défaut est `disabled`. Pour activer le profilage, définissez la valeur sur `enabled`.
   - `profiler_threshold_ms`— Quand`profiler`est défini sur`enabled`, toutes les commandes qui prennent plus de temps que `profiler-threshold-ms`sont connectés à CloudWatch. Les valeurs autorisées sont `[50-INT_MAX]`. La valeur par défaut est `100`.
   - `profiler_sampling_rate`— La fraction des opérations lentes qui doivent être profilées ou enregistrées. Les valeurs autorisées sont `[0.0-1.0]`. La valeur par défaut est `1.0`.
3. Modifiez votre cluster pour utiliser le groupe de paramètres de cluster personnalisé et configurez les exportations du journal du profileur pour qu'elles soient publiées sur Amazon CloudWatch.

   a. Dans le panneau de navigation, choisissez Clusters pour ajouter votre groupe de paramètres personnalisé à un cluster.

   b. Cliquez sur le bouton situé à gauche du nom du cluster auquel vous souhaitez associer votre groupe de paramètres. Sélectionnez Actions, puis Modify (Modifier) pour modifier votre cluster.

c. Sous Cluster options (Options de cluster), choisissez le groupe de paramètres personnalisé à partir de l'étape ci-dessus pour l'ajouter à votre cluster.

d. En dessousExportations de journaux, sélectionnezJournaux du profileurpour publier sur Amazon CloudWatch.

e. Sélectionnez Continuer pour afficher un résumé de vos modifications.

f. Après vérification de vos modifications, vous pouvez les appliquer immédiatement ou au cours de la fenêtre de maintenance suivante sous Scheduling of modifications (Planification des modifications).

g. Choisissez Modify cluster (Modifier le cluster) pour mettre à jour votre cluster avec votre nouveau groupe de paramètres.

Using the AWS CLI

La procédure suivante active le profileur sur toutes les opérations prises en charge pour le cluster `sample-cluster`.

1. Avant de commencer, vérifiez qu'un groupe de paramètres de cluster personnalisé est disponible en exécutant la commande suivante et en examinant la sortie d'un groupe de paramètres de cluster qui ne contient pas la valeur `default` dans son nom et qui comprend `docdb3.6` comme famille de groupes de paramètres. Si vous ne disposez pas d'un groupe de paramètres de cluster autre que par défaut, veuillez consulter Création de groupes de paramètres de cluster Amazon DocumentDB.

```
aws docdb describe-db-cluster-parameter-groups \
    --query 'DBClusterParameterGroups[*].
[DBClusterParameterGroupName,DBParameterGroupFamily]'
```

Dans la sortie suivante, seul `sample-parameter-group` répond à ces deux critères.

```
[
       [
           "default.docdb3.6",
           "docdb3.6"
       ],
       [
           "sample-parameter-group",
           "docdb3.6"
       ]
```

```
]
```

2. À l'aide de votre groupe de paramètres de cluster personnalisé, modifiez les paramètres suivants :

   - `profiler`— Active ou désactive le profilage des requêtes. Les valeurs autorisées sont `enabled` et `disabled`. La valeur par défaut est `disabled`. Pour activer le profilage, définissez la valeur sur `enabled`.
   - `profiler_threshold_ms`— Quand`profiler`est défini sur`enabled`, toutes les commandes prennent plus de`profiler -threshold-ms`sont connectés à CloudWatch. Les valeurs autorisées sont `[0-INT_MAX]`. La définition de cette valeur sur `0` profile toutes les opérations prises en charge. La valeur par défaut est `100`.
   - `profiler_sampling_rate`— La fraction des opérations lentes qui doivent être profilées ou enregistrées. Les valeurs autorisées sont `[0.0-1.0]`. La valeur par défaut est `1.0`.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group \
    --parameters
 ParameterName=profiler,ParameterValue=enabled,ApplyMethod=immediate \

 ParameterName=profiler_threshold_ms,ParameterValue=100,ApplyMethod=immediate \

 ParameterName=profiler_sampling_rate,ParameterValue=0.5,ApplyMethod=immediate
```

3. Modifiez votre cluster Amazon DocumentDB afin qu'il utilise`sample-parameter-group`groupe de paramètres de cluster personnalisé à partir de l'étape précédente et définit le paramètre`--enable-cloudwatch-logs-exports`pour`profiler`.

   Le code suivant modifie le cluster`sample-cluster`pour utiliser le`sample-parameter-group`à partir de l'étape précédente, et ajoute`profiler`vers les personnes activées CloudWatch Exportations de journaux.

```
aws docdb modify-db-cluster \
       --db-cluster-identifier sample-cluster \
       --db-cluster-parameter-group-name sample-parameter-group \
       --cloudwatch-logs-export-configuration '{"EnableLogTypes":["profiler"]}'
```

   Le résultat de cette opération ressemble à ceci.

```
{
    "DBCluster": {
        "AvailabilityZones": [
            "us-east-1c",
            "us-east-1b",
            "us-east-1a"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "sample-cluster",
        "DBClusterParameterGroup": "sample-parameter-group",
        "DBSubnetGroup": "default",
        "Status": "available",
        "EarliestRestorableTime": "2020-04-07T02:05:12.479Z",
        "Endpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "ReaderEndpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "MultiAZ": false,
        "Engine": "docdb",
        "EngineVersion": "3.6.0",
        "LatestRestorableTime": "2020-04-08T22:08:59.317Z",
        "Port": 27017,
        "MasterUsername": "test",
        "PreferredBackupWindow": "02:00-02:30",
        "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
        "DBClusterMembers": [
            {
                "DBInstanceIdentifier": "sample-instance-1",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            },
            {
                "DBInstanceIdentifier": "sample-instance-2",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
            }
        ],
```

```
        "HostedZoneId": "ABCDEFGHIJKLM",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-
cluster",
        "AssociatedRoles": [],
        "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
        "EnabledCloudwatchLogsExports": [
            "profiler"
        ],
        "DeletionProtection": true
    }
}
```

Désactivation du profileur Amazon DocumentDB

Pour désactiver le profileur, vous devez désactiver à la foisprofilerparamètre et exportation deprofilerse connecte à CloudWatch Journaux.

Désactivation du profileur

Vous pouvez désactiver le paramètre `profiler` à l'aide d'AWS Management Console ou de l'AWS CLI, comme suit.

Using the AWS Management Console

La procédure suivante utiliseAWS Management Consolepour désactiver Amazon DocumentDB`profiler`.

1. Connectez-vous auAWS Management Console, et ouvrez la console Amazon DocumentDB à l'adressehttps://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Groupes de paramètres. Choisissez ensuite le nom du groupe de paramètres de cluster sur lequel vous souhaitez désactiver le profileur.
3. Sur la page Cluster parameters (Paramètres de cluster) qui s'affiche, sélectionnez le bouton situé à gauche du paramètre `profiler` et choisissez Modifier.
4. Dans la boîte de dialogue Modify profiler (Modifier le profileur), choisissez `disabled` dans la liste.
5. Choisissez Modifier le paramètre de cluster.

Using the AWS CLI

Pour désactiver `profiler` sur un cluster à l'aide de l'AWS CLI, modifiez le cluster comme illustré ci-dessous.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group \
    --parameters
 ParameterName=profiler,ParameterValue=disabled,ApplyMethod=immediate
```

Désactivation de l'exportation des journaux du profileur

Vous pouvez désactiver l'exportation`profiler`se connecte à CloudWatch Se connecte en utilisant l'un ou l'autreAWS Management ConsoleouAWS CLI, comme suit.

Using the AWS Management Console

La procédure suivante utiliseAWS Management Consolepour désactiver l'exportation des journaux par Amazon DocumentDB vers CloudWatch.

1. Ouvrez la console Amazon DocumentDB à l'adressehttps://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Clusters. Choisissez le bouton à gauche du nom du cluster pour lequel vous souhaitez désactiver l'exportation des journaux.
3. Dans le menu Actions, choisissez Modify (Modifier).
4. Faites défiler la page jusqu'à la section Log exports (Exportations de journaux), puis choisissez Profiler logs (Journaux du profileur).
5. Choisissez Continue (Continuer).
6. Vérifiez vos modifications, puis choisissez quand cette modification devra être appliquée à votre cluster :
   - Appliquer pendant la fenêtre de maintenance planifiée suivante
   - Appliquer immédiatement
7. Choisissez Modifier le cluster.

Using the AWS CLI

Le code suivant modifie le cluster`sample-cluster` et désactive CloudWatch journaux du profileur.

Example

Pour Linux, macOS ou Unix :

```
aws docdb modify-db-cluster \
   --db-cluster-identifier sample-cluster \
   --cloudwatch-logs-export-configuration '{"DisableLogTypes":["profiler"]}'
```

Pour Windows :

```
aws docdb modify-db-cluster ^
   --db-cluster-identifier sample-cluster ^
   --cloudwatch-logs-export-configuration '{"DisableLogTypes":["profiler"]}'
```

Le résultat de cette opération ressemble à ceci.

```
{
    "DBCluster": {
        "AvailabilityZones": [
            "us-east-1c",
            "us-east-1b",
            "us-east-1a"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "sample-cluster",
        "DBClusterParameterGroup": "sample-parameter-group",
        "DBSubnetGroup": "default",
        "Status": "available",
        "EarliestRestorableTime": "2020-04-08T02:05:17.266Z",
        "Endpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "ReaderEndpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "MultiAZ": false,
        "Engine": "docdb",
        "EngineVersion": "3.6.0",
        "LatestRestorableTime": "2020-04-09T05:14:44.356Z",
        "Port": 27017,
        "MasterUsername": "test",
        "PreferredBackupWindow": "02:00-02:30",
        "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
        "DBClusterMembers": [
            {
                "DBInstanceIdentifier": "sample-instance-1",
```

```
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            },
            {
                "DBInstanceIdentifier": "sample-instance-2",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
            }
        ],
        "HostedZoneId": "ABCDEFGHIJKLM",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-cluster",
        "AssociatedRoles": [],
        "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
        "DeletionProtection": true
    }
}
```

Accès à vos journaux Amazon DocumentDB Profiler

Suivez ces étapes pour accéder aux journaux de votre profil sur Amazon CloudWatch.

1. Ouvrez le CloudWatch console àhttps://console.aws.amazon.com/cloudwatch/.
2. Assurez-vous que vous vous trouvez dans la même région que votre cluster Amazon DocumentDB.
3. Dans le panneau de navigation, sélectionnez Logs (Journaux).
4. Pour rechercher les journaux de profileur de votre cluster, dans la liste, choisissez `/aws/docdb/`*`yourClusterName`*`/profiler`.

Les événements d'audit pour chacune de vos instances sont disponibles sous chacun des noms d'instance respectifs.

Requêtes courantes

Voici quelques requêtes courantes que vous pouvez utiliser pour analyser vos commandes profilées. Pour plus d'informations sur CloudWatch Logs Insights, voirAnalyse des données du journal avec CloudWatch Informations sur les journauxetExemples de requêtes.

Obtenir les 10 opérations les plus lentes sur une collection spécifiée

```
filter ns="test.foo" | sort millis desc | limit 10
```

Obtenir toutes les opérations de mise à jour sur une collection qui ont duré plus de 60 ms

```
filter millis > 60 and op = "update"
```

Obtenir les 10 opérations les plus lentes du mois dernier

```
sort millis desc | limit 10
```

Obtenir toutes les requêtes avec un récapitulatif du plan COLLSCAN

```
filter planSummary="COLLSCAN"
```

Surveillance avec Performance Insights

Performance Insights complète les fonctionnalités de surveillance existantes d'Amazon DocumentDB pour illustrer les performances de votre cluster et vous aider à analyser les problèmes qui l'affectent. Le tableau de bord Performance Insights vous permet de visualiser le chargement de la base de données et de filtrer le chargement en fonction des temps d'attente, des instructions de requête, des hôtes ou des applications.

> **Note**
>
> Performance Insights est uniquement disponible pour les clusters basés sur des instances Amazon DocumentDB 3.6, 4.0 et 5.0.

En quoi est-ce utile ?

- Visualisez les performances de la base de données : visualisez la charge pour déterminer quand et où elle se trouve sur la base de données
- Déterminer la cause de la charge sur la base de données : déterminez les requêtes, les hôtes et les applications qui contribuent à la charge sur votre instance
- Déterminez à quel moment votre base de données est chargée : zoomez sur le tableau de bord Performance Insights pour vous concentrer sur des événements spécifiques ou effectuez un zoom arrière pour examiner les tendances sur une période plus longue
- Alerte concernant le chargement de la base de données : accédez automatiquement aux nouvelles mesures de charge de base de données, à partir CloudWatch desquelles vous pouvez surveiller les mesures de charge de la base de données ainsi que d'autres mesures DocumentDB et définir des alertes à leur sujet

Quelles sont les limites d'Amazon DocumentDB Performance Insights ?

- Les Performance Insights dans la région AWS GovCloud (ouest des États-Unis) ne sont pas encore disponibles
- Performance Insights for DocumentDB conserve jusqu'à 7 jours de données de performance
- Les requêtes de plus de 1 024 Ko ne sont pas agrégées dans Performance Insights

Rubriques

- Performance Insights pour les contre-métriques

Concepts relatifs aux Performances Insights

Rubriques

- Sessions actives en moyenne
- Dimensions
- Nombre maximal de vCPU

Sessions actives en moyenne

Database load (DB load) (Charge de la base de données, ou charge DB) mesure le niveau d'activité de votre base de données. La métrique clé de Performance Insights est `DB Load`, qui est collectée toutes les secondes. L'unité de la `DBLoad` métrique est le nombre moyen de sessions actives (AAS) pour une instance DocumentDB.

Une session active est une connexion qui a soumis un travail à l'instance DocumentDB et qui attend une réponse. Par exemple, si vous soumettez une requête à une instance DocumentDB, la session de base de données est active pendant que l'instance traite la requête.

Pour obtenir les sessions actives en moyenne, Performance Insights échantillonne le nombre de sessions exécutant simultanément une requête. L'AAS correspond au nombre total de sessions divisé par le nombre total d'échantillons. Le tableau suivant présente cinq exemples consécutifs d'une requête en cours d'exécution.

Exemple	Nombre de sessions exécutant la requête	AAS	Calcul
1	2	2	2 sessions / 1 échantillon
2	0	1	2 sessions / 2 échantillons
3	4	2	6 sessions / 3 échantillons

Exemple	Nombre de sessions exécutant la requête	AAS	Calcul
4	0	1.5	6 sessions / 4 échantillons
5	4	2	10 sessions / 5 échantillons

Dans l'exemple précédent, la charge de base de données pour l'intervalle de temps compris entre 1 et 5 est de 2 AAS. Une augmentation de la charge de base de données signifie que, en moyenne, un plus grand nombre de sessions s'exécutent sur la base de données.

Dimensions

La métrique `DB Load` est différente des autres métriques de série chronologique, car vous pouvez la décomposer en sous-composants appelés dimensions. Vous pouvez considérer les dimensions comme des catégories pour les différentes caractéristiques de la métrique `DB Load`. Lorsque vous diagnostiquez des problèmes de performances, les dimensions les plus utiles sont les états d'attente et la première requête.

états d'attente

Un état d'attente oblige une instruction de requête à attendre qu'un événement spécifique se produise avant de pouvoir continuer à s'exécuter. Par exemple, une instruction de requête peut attendre qu'une ressource verrouillée soit déverrouillée. En combinant les états `DB Load` d'attente, vous pouvez obtenir une image complète de l'état de la session. Voici les différents états d'attente de DocumentDB :

État d'attente de DocumentDB	Description de l'état d'attente
Loquet	L'état d'attente Latch se produit lorsque la session attend de mettre en page le pool de mémoire tampon. Les entrées et sorties fréquentes du pool de mémoire tampon peuvent se produire plus souvent lorsque le système traite fréquemment des requêtes volumineuses, que des collections sont

État d'attente de DocumentDB	Description de l'état d'attente
	scannées ou lorsque le pool de mémoire tampon est trop petit pour gérer l'ensemble de travail.
CPU	L'état d'attente du processeur se produit lorsque la session est en attente sur le processeur.
CollectionLock	L'état d' CollectionLock attente se produit lorsque la session attend de verrouiller la collection. Ces événements se produisent lorsque des opérations DDL sont effectuées sur la collection.
DocumentLock	L'état d' DocumentLock attente se produit lorsque la session attend d'obtenir un verrou sur un document. Un nombre élevé d'écritures simultanées sur le même document contribue ra à augmenter le nombre d'états d' DocumentL ockattente sur ce document.
SystemLock	L'état d' SystemLock attente se produit lorsque la session est en attente sur le système. Cela peut se produire en cas de requêtes fréquentes et longues, de transactions de longue durée ou de forte simultanéité sur le système.
E/S	L'état d'attente d'E/S se produit lorsque la session en attente d'E/S est terminée.
BufferLock	L'état d' BufferLock attente se produit lorsque la session attend d'obtenir un verrou sur une page partagée dans la mémoire tampon. BufferLoc kles états d'attente peuvent être prolongés si d'autres processus maintiennent des curseurs ouverts sur les pages demandées.

État d'attente de DocumentDB	Description de l'état d'attente
LowMemThrottle	L'état d' LowMemThrottle attente se produit lorsque la session est en attente en raison d'une forte pression de mémoire sur l'instanc e Amazon DocumentDB. Si cet état persiste pendant une longue période, envisagez de redimensionner l'instance pour fournir de la mémoire supplémentaire. Pour plus d'informa tions, consultez Resource Governor.
BackgroundActivity	L'état d' BackgroundActivity attente se produit lorsque la session attend des processus internes du système.
Autre	L'autre état d'attente est un état d'attente interne. Si cet état persiste pendant une longue période, pensez à mettre fin à cette requête. Pour plus d'informations, voir Comment rechercher et mettre fin aux requêtes bloquées ou de longue durée ?

Requêtes les plus fréquentes

Alors que les états d'attente indiquent les goulots d'étranglement, les requêtes les plus fréquentes indiquent les requêtes qui contribuent le plus à la charge de la base de données. Par exemple, de nombreuses requêtes peuvent être en cours d'exécution sur la base de données, mais une seule d'entre elles peut consommer 99 % de la charge de la base de données. Dans ce cas, la charge élevée peut indiquer un problème avec la requête.

Nombre maximal de vCPU

Dans le tableau de bord, le graphique Database load (Charge de base de données)collecte, regroupe et affiche les informations de session. Pour voir si les sessions actives dépassent l'utilisation maximale de l'UC, examinez leur relation sur la ligne Max vCPU (UC virtuelle max). La valeur maximale de vCPU est déterminée par le nombre de cœurs de vCPU (processeur virtuel) pour votre instance DocumentDB.

Si la charge de la base de données est souvent au-dessus de la ligne Max vCPU (UC virtuelle max)et que l'état d'attente principal est CPU, cela signifie que l'UC est surchargée. Dans ce cas, vous souhaiterez peut-être limiter les connexions à l'instance, régler les requêtes impliquant une charge CPU élevée ou envisager une classe d'instance plus importante. Quel que soit leur état d'attente, les instances élevées et régulières indiquent que des problèmes de goulots d'étranglement ou de conflits de ressources devront peut-être être résolus. Cela peut être vrai même si la charge de la base de données ne dépasse pas la ligne Max vCPU (UC virtuelle max).

Activation et désactivation de Performance Insights

Pour utiliser Performance Insights, vous devez l'activer sur votre instance de base de données. Vous pourrez le désactiver ultérieurement si nécessaire. L'activation et la désactivation de Performance Insights ne provoquent pas de temps d'arrêt, de redémarrage ou de basculement.

L'agent Performance Insights consomme une quantité limitée d'UC et de mémoire sur l'hôte de base de données. Lorsque la charge de base de données est élevée, l'agent limite l'impact sur les performances en collectant des données moins fréquemment.

Activation de Performance Insights lors de la création d'un cluster

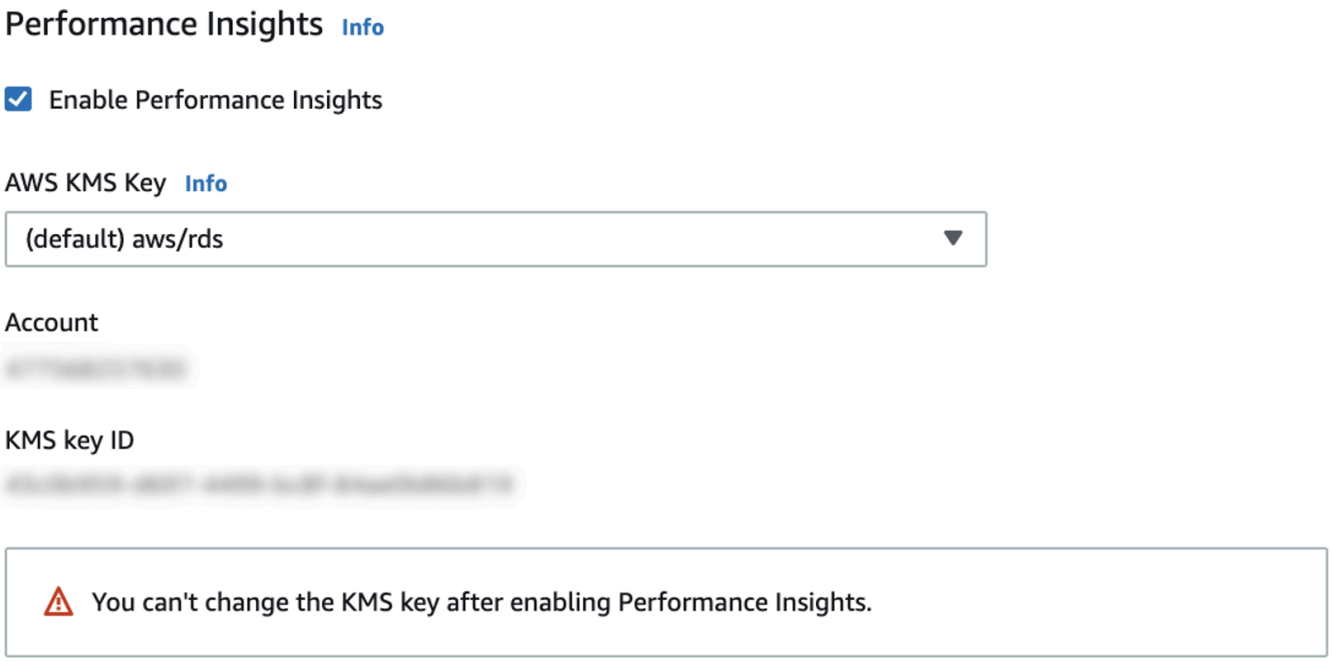
Dans la console, vous pouvez activer ou désactiver Performance Insights lorsque vous créez ou modifiez une nouvelle instance de base de données.

Utilisation du AWS Management Console

Dans la console, vous pouvez activer Performance Insights lorsque vous créez un cluster DocumentDB. Lorsque vous créez un nouveau cluster DocumentDB, activez Performance Insights en choisissant Enable Performance Insights dans la section Performance Insights.

Instructions relatives à la console

1. Pour créer un cluster, suivez les instructions de création d'un cluster Amazon DocumentDB.
2. Sélectionnez Activer Performance Insights dans la section Performance Insights.



> **Note**
>
> La période de conservation des données de Performance Insights sera de sept jours.

AWS KMSclé — Spécifiez votre clé AWS KMS. Performance Insights chiffre toutes les données potentiellement sensibles à l'aide votre clé AWS KMS. Les données sont chiffrées en transit et au repos. Pour plus d'informations, consultez Configuration d'une AWS AWS KMS politique pour Performance Insights.

Activation et désactivation lors de la modification d'une instance

Vous pouvez modifier une instance de base de données pour activer ou désactiver Performance Insights à l'aide de la console ouAWS CLI.

Using the AWS Management Console

Instructions relatives à la console

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Choisissez Clusters.

3. Choisissez une instance de base de données, puis Modifiez.
4. Dans la section Performance Insights, choisissez Enable Performance Insights ou Disable Performance Insights.

> **Note**
>
> Si vous choisissez Enable Performance Insights, vous pouvez spécifier votre AWS AWS KMS clé. Performance Insights chiffre toutes les données potentiellement sensibles à l'aide votre clé AWS KMS. Les données sont chiffrées en transit et au repos. Pour plus d'informations, consultez la section Chiffrement des données Amazon DocumentDB au repos.

5. Choisissez Continuer.
6. Pour Scheduling of Modifications (Planification des modifications), choisissez Appliquer immédiatement. Si vous choisissez Appliquer lors de la prochaine fenêtre de maintenance planifiée, votre instance ignore ce paramètre et active immédiatement Performance Insights.
7. Choisissez Modify instance (Modifier l'instance).

Using the AWS CLI

Lorsque vous utilisez les `modify-db-instance` AWS AWS CLI commandes `create-db-instance` ou, vous pouvez activer Performance Insights en spécifiant`--enable-performance-insights`, ou le désactiver en spécifiant`--no-enable-performance-insights`.

La procédure suivante décrit comment activer ou désactiver Performance Insights pour une instance de base de données à l'aide du AWSAWS CLI.

AWSAWS CLIinstructions

Appelez la `modify-db-instance` AWS AWS CLI commande et fournissez les valeurs suivantes :

- `--db-instance-identifer`— Le nom de l'instance de base de données
- `--enable-performance-insights` pour activer ou `--no-enable-performance-insights` pour désactiver

Example

L'exemple suivant active Performance Insights pour `sample-db-instance` :

For Linux, macOS, or Unix:

```
aws docdb modify-db-instance \
    --db-instance-identifier sample-db-instance \
    --enable-performance-insights
```

For Windows:

```
aws docdb modify-db-instance ^
    --db-instance-identifier sample-db-instance ^
    --enable-performance-insights
```

Configuration des politiques d'accès pour Performance Insights

Pour accéder à Performance Insights, vous devez disposer des autorisations appropriées d'AWS Identity and Access Management (IAM). Vous disposez des options suivantes pour accorder l'accès :

- Attachez la politique gérée par `AmazonRDSPerformanceInsightsReadOnly` à un jeu d'autorisations ou à un rôle.
- Créez une politique IAM personnalisée et attachez-la à un jeu d'autorisations ou à un rôle.

En outre, si vous avez spécifié une clé gérée par le client lorsque vous avez activé Performance Insights, assurez-vous que les utilisateurs de votre compte disposent des autorisations `kms:Decrypt` et `kms:GenerateDataKey` sur la clé KMS.

> **Note**
>
> Pour la gestion encryption-at-rest des AWS KMS clés et des groupes de sécurité, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon RDS.

Associer la PerformanceInsightsReadOnly politique Amazon RDS à un IAM principal

`AmazonRDSPerformanceInsightsReadOnly`est une politique AWS gérée qui donne accès à toutes les opérations en lecture seule de l'API Performance Insights d'Amazon DocumentDB.

Actuellement, toutes les opérations de cette API sont en lecture seule. Si vous attachez `AmazonRDSPerformanceInsightsReadOnly` à un jeu d'autorisations ou à un rôle, le destinataire peut utiliser Performance Insights avec d'autres fonctions de la console.

Création d'une politique IAM personnalisée pour Performance Insights

Pour les utilisateurs qui ne bénéficient pas d'un accès à la politique `AmazonRDSPerformanceInsightsReadOnly`, vous pouvez accorder l'accès à Performance Insights en créant ou modifiant une politique IAM gérée par l'utilisateur. Lorsque vous associez la politique à un ensemble d'autorisations ou à un rôle, le destinataire peut utiliser Performance Insights.

Pour créer une politique personnalisée

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Policies (Politiques).
3. Sélectionnez Create policy (Créer une politique).
4. Sur la page Créer une stratégie, choisissez l'onglet JSON.
5. Copiez et collez le texte suivant, en remplaçant *`us-east-1`* par le nom de votre région AWS et *`111122223333`* par le numéro de votre compte client.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "rds:DescribeDBInstances",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "rds:DescribeDBClusters",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:DescribeDimensionKeys",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        },
        {
```

```
            "Effect": "Allow",
            "Action": "pi:GetDimensionKeyDetails",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:GetResourceMetadata",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:GetResourceMetrics",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:ListAvailableResourceDimensions",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:ListAvailableResourceMetrics",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        }
    ]
}
```

6. Choisissez Examiner une stratégie.
7. Indiquez un nom pour la stratégie et éventuellement une description, puis choisissez Créer une stratégie.

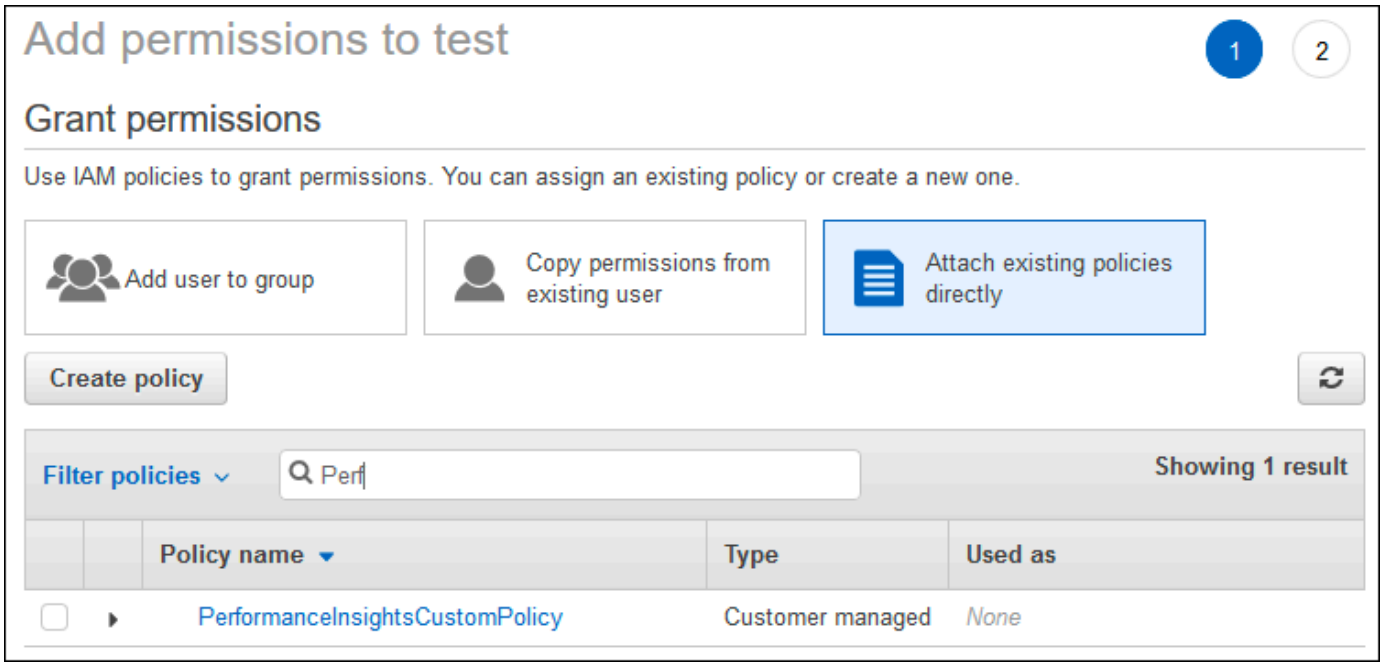
Vous pouvez désormais attacher la politique à un jeu d'autorisations ou à un rôle. La procédure suivante suppose que vous disposez déjà d'un utilisateur à cette fin.

Pour attacher la politique à un utilisateur

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation, sélectionnez Users.
3. Choisissez un utilisateur existant dans la liste.

> **Important**
>
> Pour utiliser Performance Insights, assurez-vous d'avoir accès à Amazon DocumentDB en plus de la politique personnalisée. Par exemple, la politique ReadOnlyAccess prédéfinie de base de données fournit un accès en lecture seule à Amazon AmazonDocDocDB. Pour plus d'informations, consultez Gérer l'accès à l'aide de politiques.

4. Sur la page Récapitulatif, choisissez Ajouter des autorisations.
5. Choisissez Attacher directement les stratégies existantes. Pour Recherche, tapez les premiers caractères du nom de votre stratégie, comme illustré ci-après.



6. Choisissez votre stratégie, puis sélectionnez Suivant : Vérification.
7. Choisissez Add permissions.

Configuration d'une politique AWS KMS pour Performance Insights

Performance Insights utilise une AWS KMS key pour chiffrer les données sensibles. Lorsque vous activez Performance Insights via l'API ou la console, vous disposez des options suivantes :

- Choisissez la valeur Clé gérée par AWS par défaut.

Amazon DocumentDB utilise le Clé gérée par AWS pour votre nouvelle instance de base de données. Amazon DocumentDB crée un Clé gérée par AWS pour votre AWS compte. Votre AWS compte possède un identifiant Amazon DocumentDB différent Clé gérée par AWS pour chaque AWS région.

- Choisissez une clé gérée par le client.

  Si vous spécifiez une clé gérée par le client, les utilisateurs de votre compte qui appellent l'API Performance Insights ont besoin des autorisations `kms:Decrypt` et `kms:GenerateDataKey` sur la clé KMS. Vous pouvez configurer ces autorisations via des politiques IAM. Toutefois, nous vous recommandons de gérer ces autorisations via votre politique de clé KMS. Pour plus d'informations, veuillez consulter Utilisation des stratégies de clé dans AWS KMS.

Example

L'exemple de politique de clé suivant montre comment ajouter des instructions à votre politique KMS. Ces instructions permettent d'accéder à Performance Insights. Selon la manière dont vous utilisez leAWS KMS, vous souhaiterez peut-être modifier certaines restrictions. Avant d'ajouter des instructions à votre politique, supprimez tous les commentaires.

```
{
 "Version" : "2012-10-17",
 "Id" : "your-policy",
 "Statement" : [ {
    //This represents a statement that currently exists in your policy.
 }
 ....,
 //Starting here, add new statement to your policy for Performance Insights.
 //We recommend that you add one new statement for every RDS/DocumentDB instance
 {
    "Sid" : "Allow viewing RDS Performance Insights",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            //One or more principals allowed to access Performance Insights
            "arn:aws:iam::444455556666:role/Role1"
        ]
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
```

```
    ],
    "Resource": "*",
    "Condition" :{
        "StringEquals" : {
            //Restrict access to only RDS APIs (including Performance Insights).
            //Replace *region* with your AWS Region.
            //For example, specify us-west-2.
            "kms:ViaService" : "rds.*region*.amazonaws.com"
        },
        "ForAnyValue:StringEquals": {
            //Restrict access to only data encrypted by Performance Insights.
            "kms:EncryptionContext:aws:pi:service": "rds",
            "kms:EncryptionContext:service": "pi",

            //Restrict access to a specific DocDB instance.
            //The value is a DbiResourceId.
           "kms:EncryptionContext:aws:rds:db-id": "db-AAAAABBBBBCCCCDDDDDEEEEE"
        }
    }
}
```

Analyse des métriques à l'aide du tableau de bord de Performance Insights

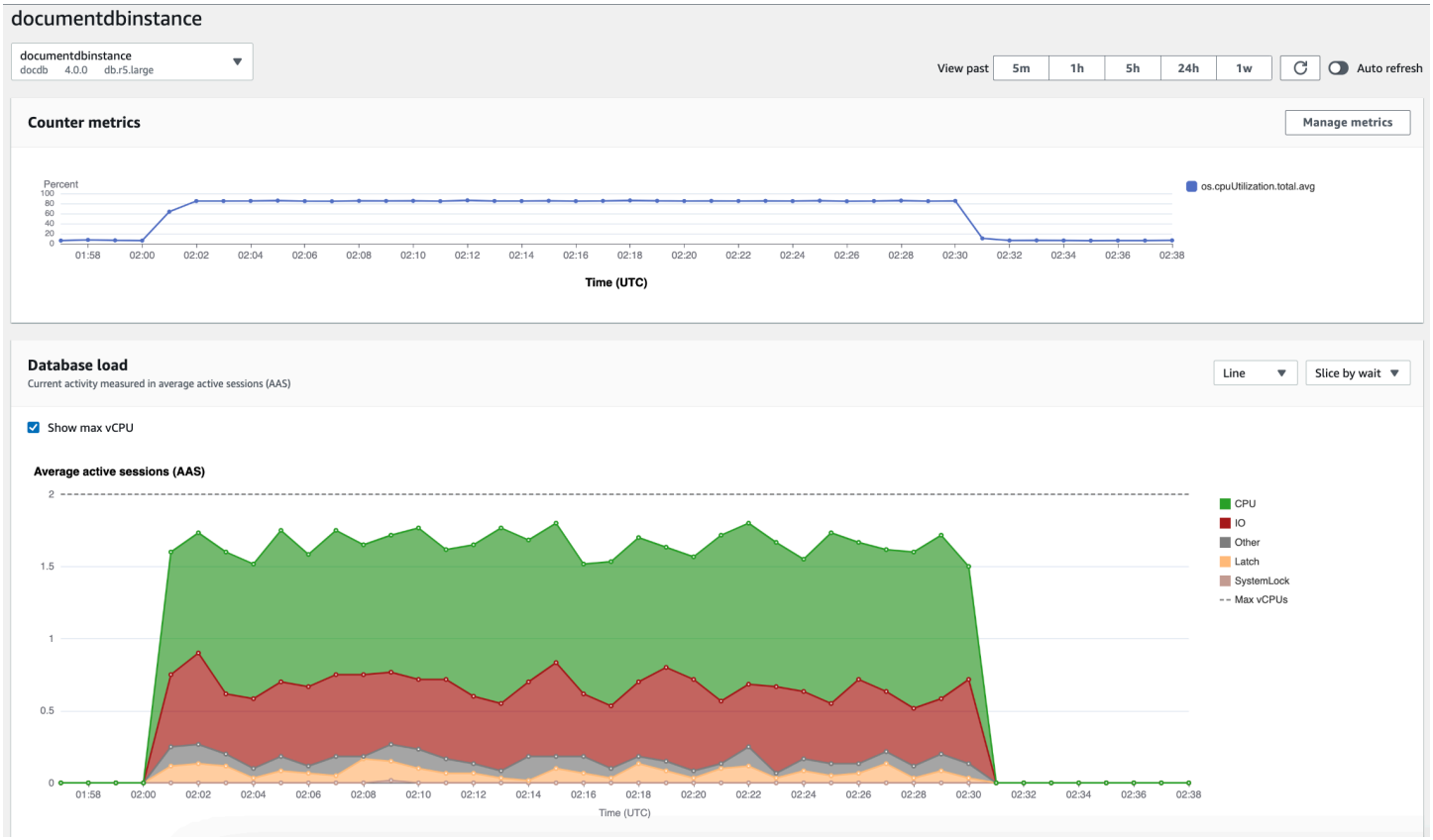
Le tableau de bord de Performance Insights contient des informations sur les performances des bases de données qui vous aideront à analyser et à résoudre les problèmes de performances. Sur la page principale du tableau de bord, vous pouvez consulter les informations relatives au chargement de la base de données (charge de base de données). Vous pouvez « découper » le chargement de la base de données en fonction de dimensions telles que les états d'attente ou les requêtes.

Rubriques

- Présentation du tableau de bord Performance Insights
- Ouverture du tableau de bord de Performance Insights
- Analyse du chargement de la base de données par états d'attente
- Vue d'ensemble de l'onglet Requêtes les plus fréquentes
- Zoomer sur le graphique de charge de la base de données

Présentation du tableau de bord Performance Insights

Le tableau de bord est le moyen le plus simple d'interagir avec Performance Insights. L'exemple suivant montre le tableau de bord d'une instance Amazon DocumentDB. Par défaut, le tableau de bord de Performance Insights affiche les données pour l'heure précédente.



Le tableau de bord est divisé entre les parties suivantes :

1. Indicateurs de comptage : affiche les données relatives à des indicateurs de performance spécifiques.
2. Charge de base de données — Indique comment la charge de base de données se compare à la capacité de l'instance de base de données, telle que représentée par la ligne Max vCPU.
3. Dimensions supérieures — Affiche les dimensions supérieures qui contribuent à la charge de la base de données. Ces dimensions incluent `waitsqueries`,`hosts`,`databases`, et`applications`.

Rubriques

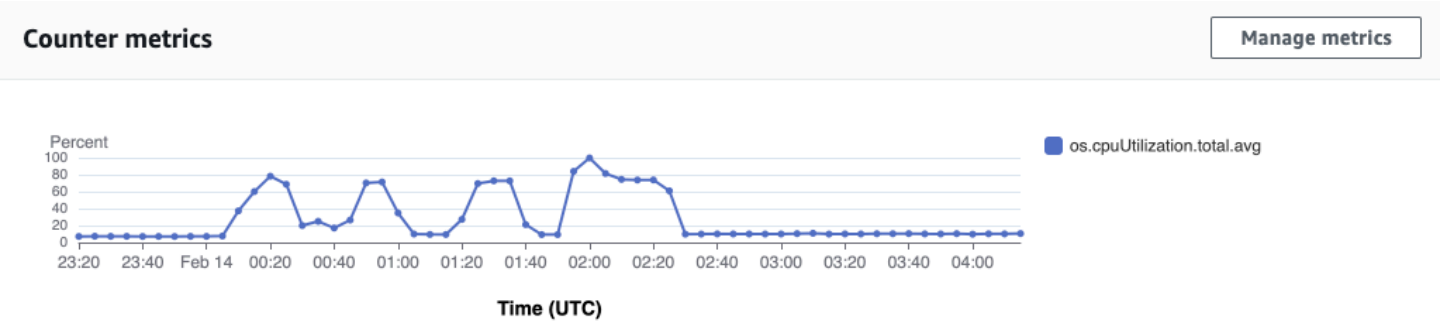
- Graphique Counter Metrics (Métriques de compteur)

- Graphique Database Load (Charge de la base de données)
- Tableau des dimensions principales

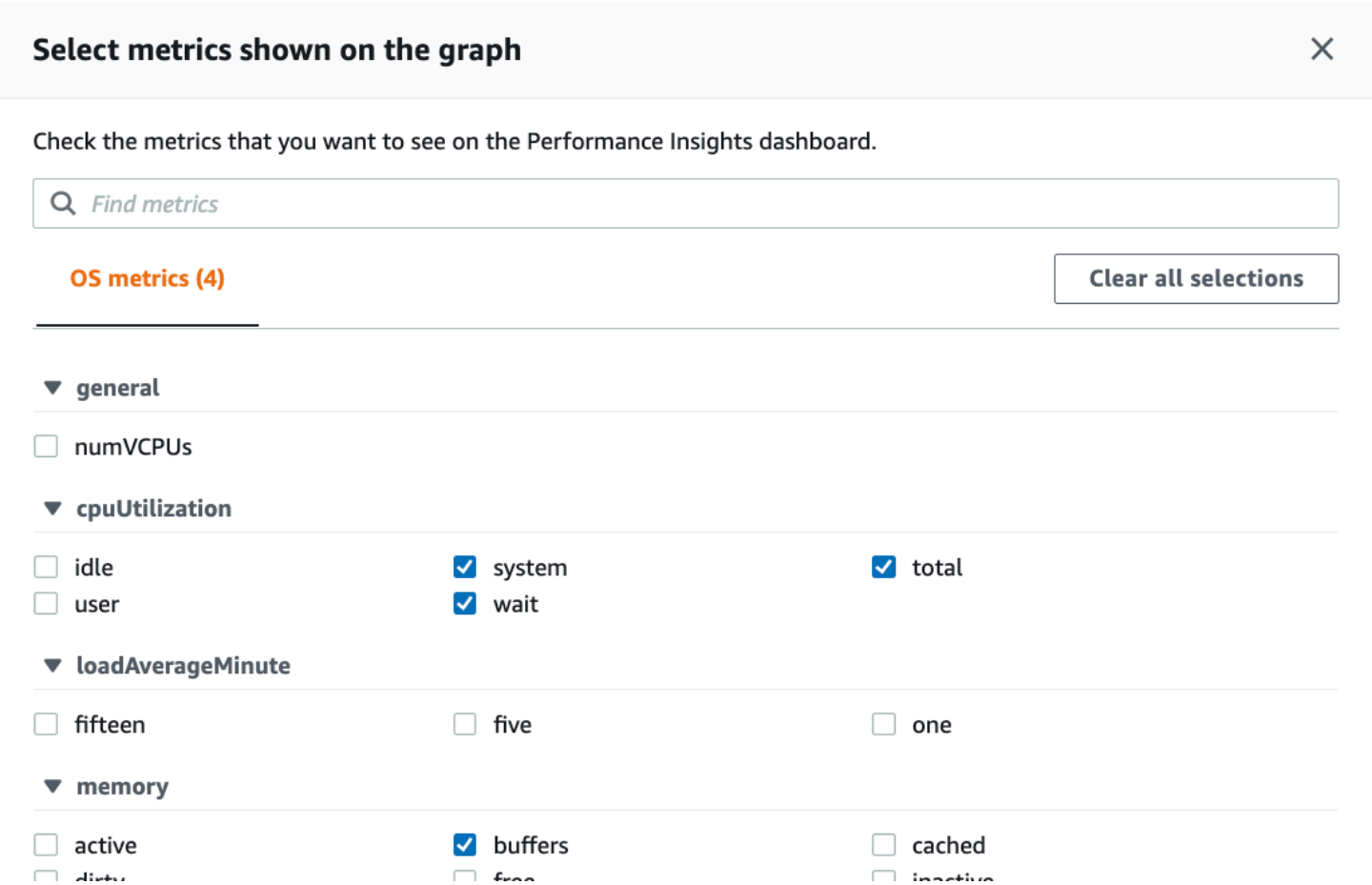
Graphique Counter Metrics (Métriques de compteur)

Grâce aux métriques de compteur, vous pouvez personnaliser le tableau de bord de Performance Insights de sorte à inclure jusqu'à 10 graphiques supplémentaires. Ces graphiques présentent une sélection de dizaines de métriques du système d'exploitation. Vous pouvez établir des corrélations entre ces informations et la charge de la base de données pour identifier et analyser les problèmes de performances.

Le graphique Counter Metrics (Métriques de compteur) affiche les données des compteurs de performances.



Pour modifier les compteurs de performance, choisissez Gérer les métriques. Vous pouvez sélectionner plusieurs métriques du système d'exploitation, comme indiqué dans la capture d'écran suivante. Pour afficher les détails relatifs à une métrique, passez la souris sur le nom de la métrique.



Graphique Database Load (Charge de la base de données)

Le graphique de charge de la base de données montre comment l'activité de la base de données se compare à la capacité de l'instance telle que représentée par la ligne Max vCPU. Par défaut, le graphique en courbes empilées représente la charge de la base de données sous forme de sessions actives en moyenne par unité de temps. La charge de la base de données est découpée (groupée) par états d'attente.

Charge de base de données tranchée par dimensions

Vous pouvez afficher la charge sous la forme de sessions actives regroupées par dimensions prises en charge. L'image suivante montre les dimensions de l'instance Amazon DocumentDB.

Détails de charge de base de données pour un élément de dimension

Pour afficher les détails d'un élément de charge de base de données dans une dimension, passez la souris sur le nom d'élément. L'image suivante montre les détails d'une instruction de requête.

Pour afficher les détails d'un élément pour la période sélectionnée dans la légende, survolez cet élément.

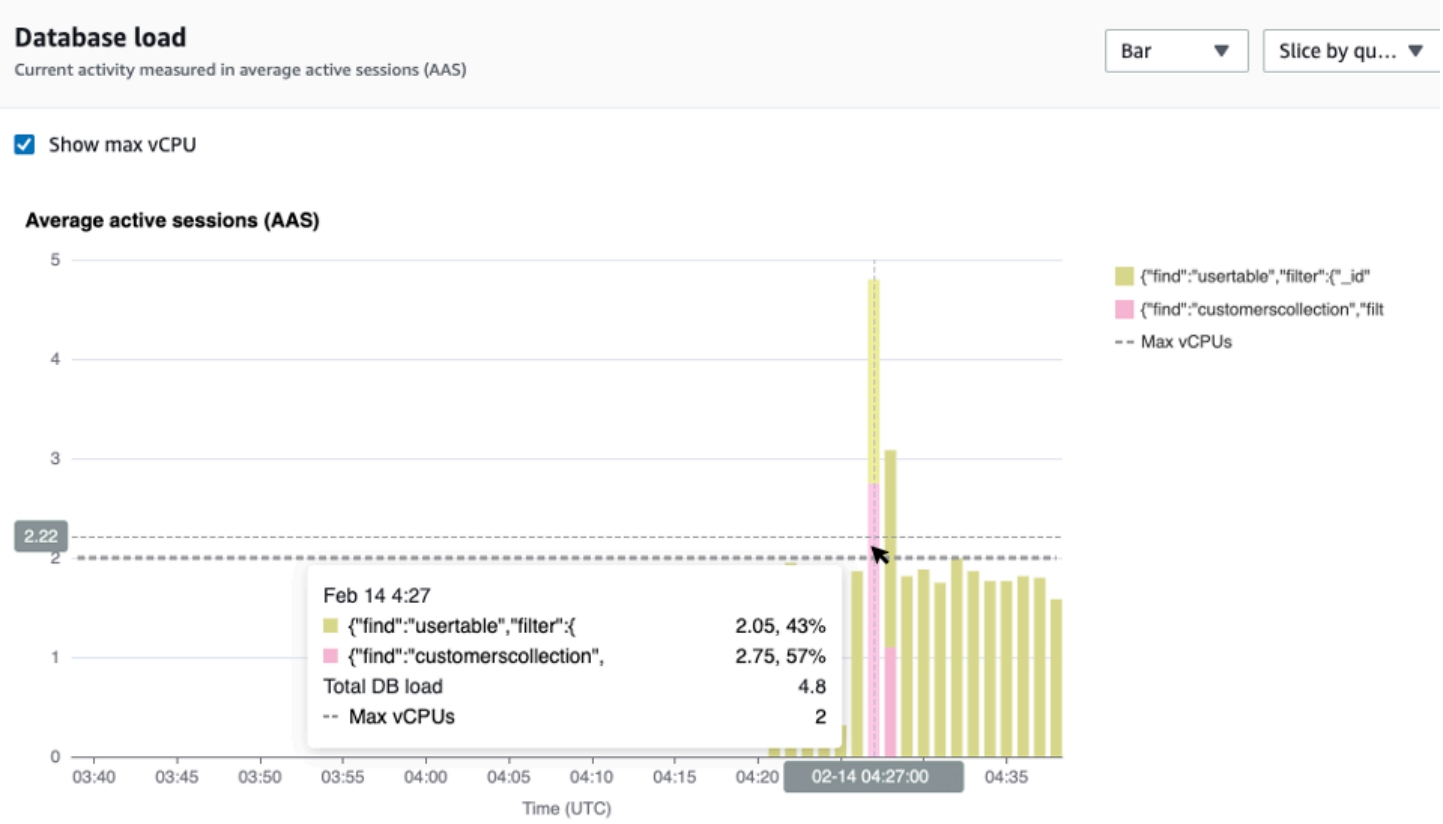
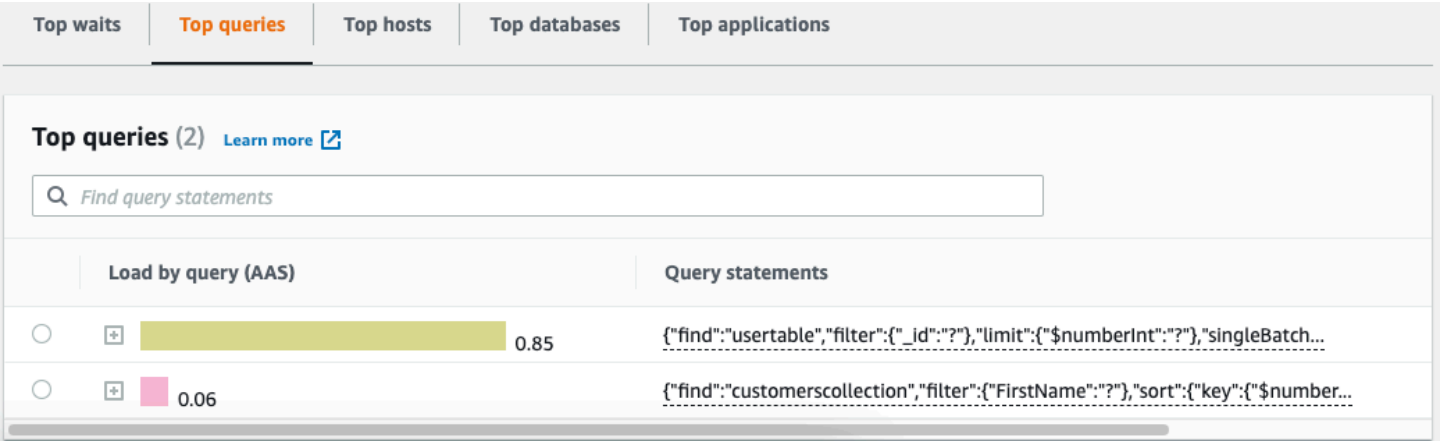


Tableau des dimensions principales

Le tableau des dimensions supérieures répartit la charge de base de données selon différentes dimensions. Une dimension est une catégorie ou « tranche » qui représente l'une des différentes caractéristiques de la charge de la base de données. Si la dimension est query, Top queries affiche les instructions de requête qui contribuent le plus à la charge de la base de données.

Choisissez l'un des onglets de dimension suivants.



Le tableau suivant fournit une brève description de chaque onglet.

Description	Description
Événements pour l'attente backend de la base de données attend	Événements pour l'attente backend de la base de données attend
Les instructions les plus fréquentes en cours d'exécution	Les instructions les plus fréquentes en cours d'exécution
Adresse IP et hôtes le port de l'hôte du client connecté	Adresse IP et hôtes le port de l'hôte du client connecté

Description
Nom de la base de données à laquelle le client est connecté
Nom de l'application connectée à la base de données

Pour savoir comment analyser les requêtes à l'aide de l'onglet Requêtes les plus fréquentes, consultezVue d'ensemble de l'onglet Requêtes les plus fréquentes.

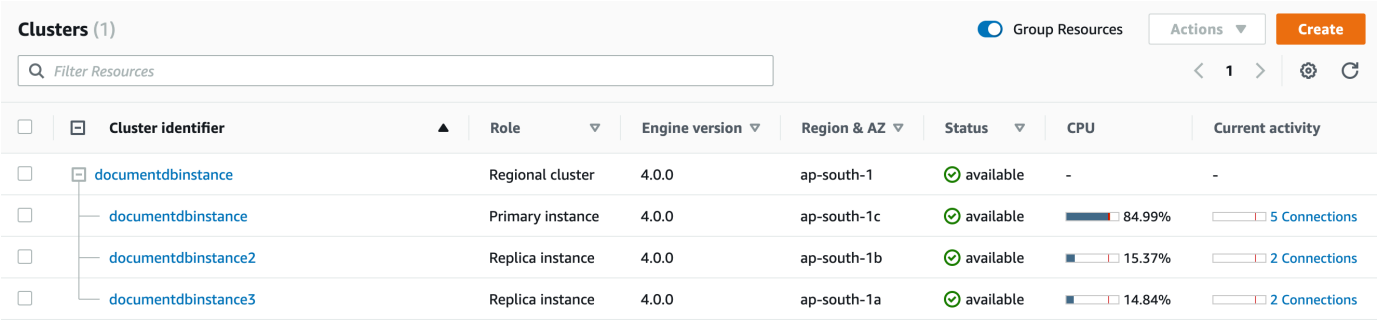
Ouverture du tableau de bord de Performance Insights

Pour consulter le tableau de bord Performance Insights dans la console de AWS gestion, procédez comme suit :

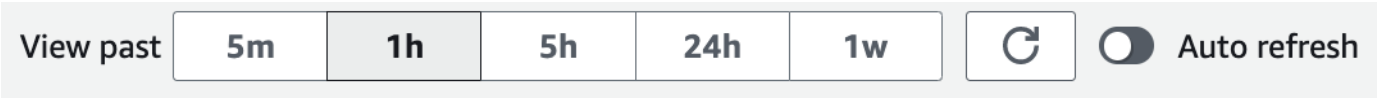
1. Ouvrez la console Performance Insights à l'adresse https://console.aws.amazon.com/docdb/.

2. Choisissez une instance de base de données. Le tableau de bord Performance Insights est affiché pour cette instance Amazon DocumentDB.

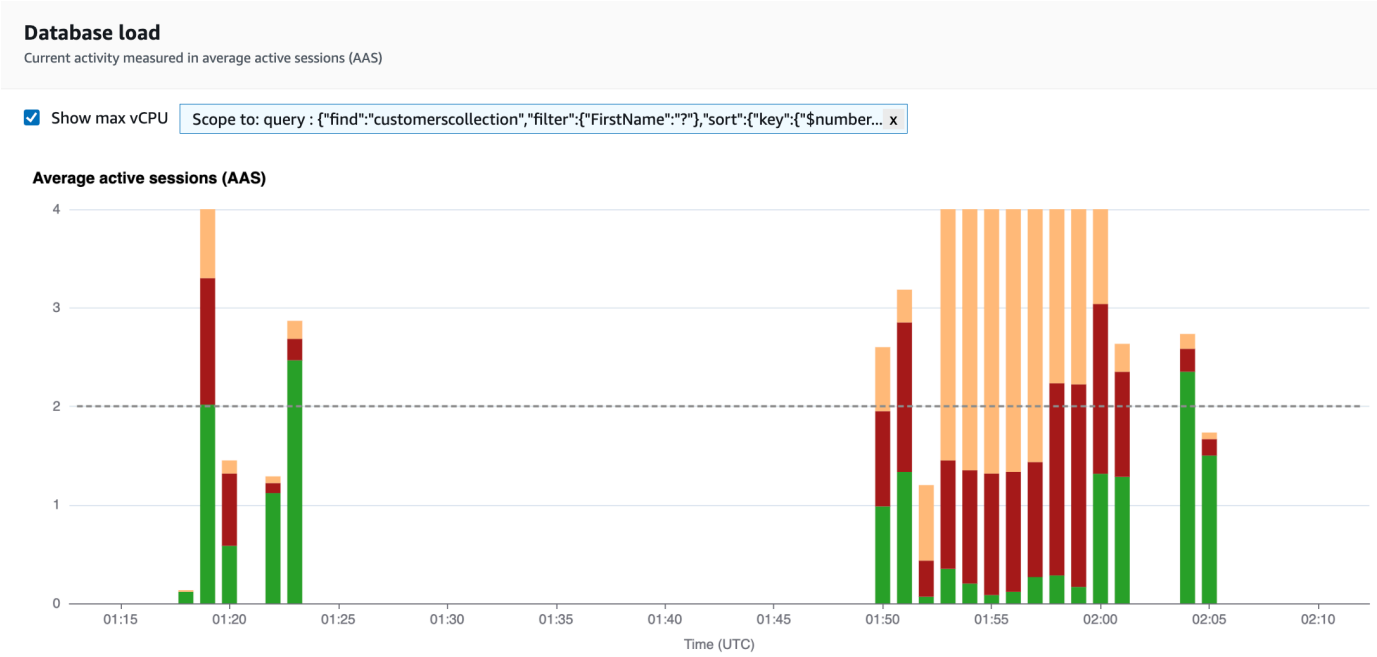
   Pour les instances Amazon DocumentDB sur lesquelles Performance Insights est activé, vous pouvez également accéder au tableau de bord en choisissant l'élément Sessions dans la liste des instances. Sous Activité actuelle, l'élément Sessions affiche la charge de base de données dans les sessions actives moyennes lors des cinq dernières minutes. La barre affiche visuellement le chargement. Lorsque la barre est vide, l'instance est inactive. La barre se remplit de bleu à mesure que le chargement augmente. Lorsque la charge dépasse le nombre de processeurs virtuels (vCPU) de la classe d'instance, la barre devient rouge, ce qui indique un goulot d'étranglement potentiel.



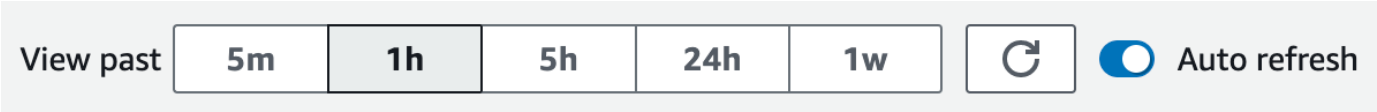
3. (Facultatif) Choisissez un intervalle de temps différent en sélectionnant un bouton dans le coin supérieur droit. Par exemple, pour modifier l'intervalle à 1 heure, sélectionnez 1 heure.



   Dans la capture d'écran suivante, l'intervalle de chargement de la base de données est de 1 heure.



4. Pour actualiser automatiquement vos données, activez l'option Actualisation automatique.



Le tableau de bord Performance Insight s'actualise automatiquement avec de nouvelles données. Le taux de rafraîchissement dépend de la quantité de données affichées :

- Si vous choisissez 5 minutes, les données seront actualisées toutes les 5 secondes.
- 1 heure se réactualise toutes les minutes.
- 5 heures se réactualise toutes les minutes.
- 24 heures se réactualise toutes les 5 minutes.
- 1 semaine se réactualise toutes les heures.

Analyse du chargement de la base de données par états d'attente

Si le graphique de charge de base de données (charge de base de données) indique un goulot d'étranglement, vous pouvez savoir d'où vient la charge. Pour ce faire, examinez le tableau des principaux éléments de charge en dessous du graphique Database load (Charge de la base de données). Choisissez un élément en particulier, comme une requête ou une application, pour accéder à cet élément et en voir les détails.

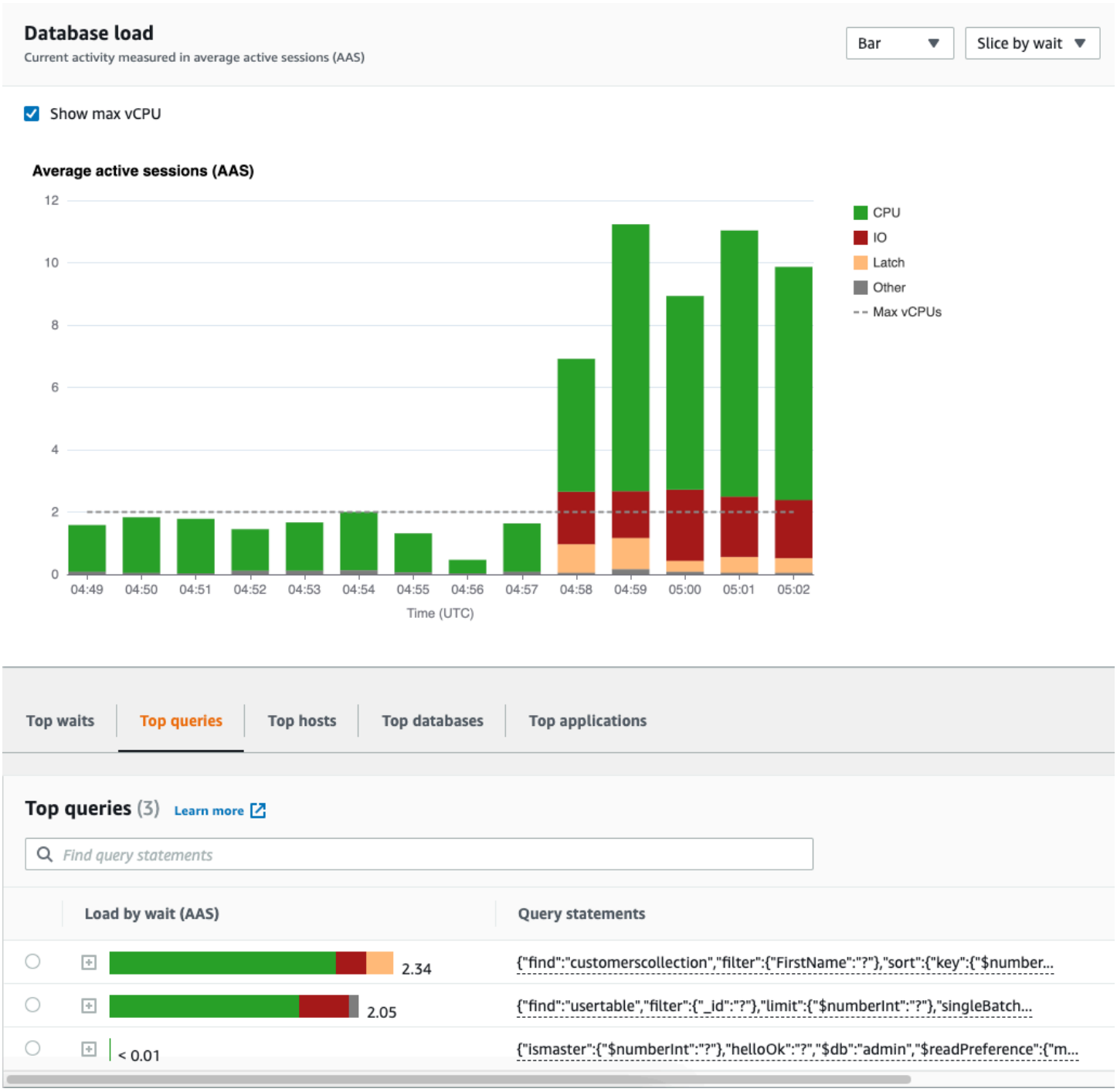
La charge de base de données regroupée par temps d'attente et par requêtes les plus fréquentes fournit généralement le meilleur aperçu des problèmes de performances. L'affichage de la charge de la base de données en fonction de l'attente indique s'il existe des goulots d'étranglement liés aux ressources ou à des actions simultanées dans la base de données. Dans ce cas, l'onglet Principales requêtes du tableau des éléments à charger le plus souvent indique quelles requêtes sont à l'origine de cette charge.

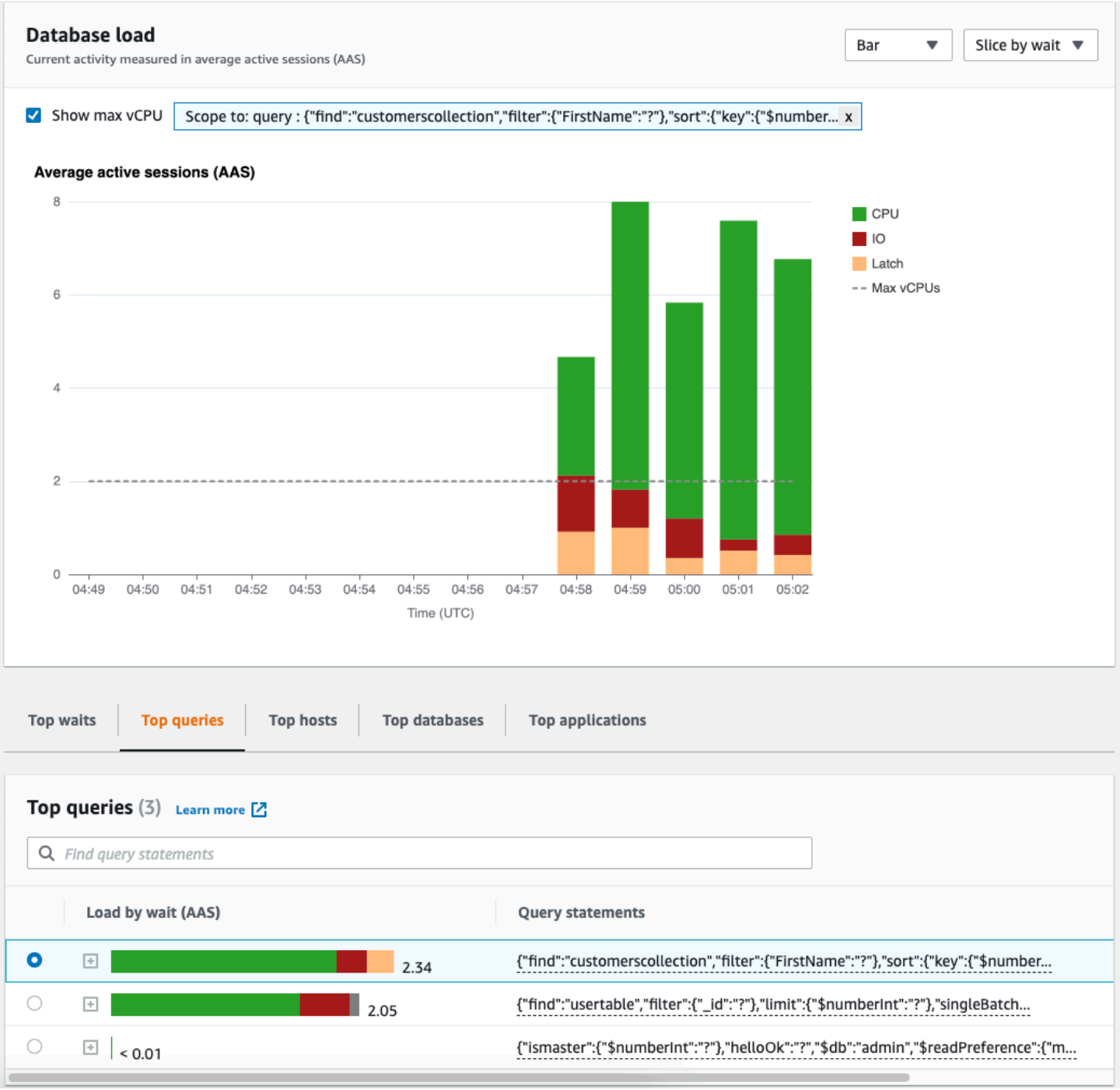
Votre flux de travail standard pour diagnostiquer les problèmes de performances se présente comme suit :

1. Dans le graphique Database load (Charge de la base de données), regardez s'il existe des incidents de charge de base de données qui dépassent la ligne Max CPU (CPU max).
2. Si c'est le cas, observez le graphique Database load (Charge de la base de données) et identifiez le ou les états d'attente qui sont les principaux responsables.
3. Identifiez les requêtes de synthèse à l'origine de la charge en identifiant les requêtes figurant dans l'onglet Principales requêtes du tableau des éléments de chargement les plus importants qui contribuent le plus à ces états d'attente. Vous pouvez les identifier à l'aide de la colonne Load by Wait (AAS).
4. Choisissez l'une de ces requêtes de synthèse dans l'onglet Principales requêtes pour la développer et voir les requêtes enfants qui la composent.

Vous pouvez également voir quels hôtes ou quelles applications contribuent le plus à la charge en sélectionnant les meilleurs hôtes ou les meilleures applications, respectivement. Les noms des applications sont spécifiés dans la chaîne de connexion à l'instance Amazon DocumentDB. `Unknown`indique que le champ d'application n'a pas été spécifié.

Par exemple, dans le tableau de bord suivant, les temps d'attente du processeur représentent la majeure partie de la charge de base de données. La sélection de la première requête sous Principales requêtes permet d'étendre le graphique de charge de la base de données afin de se concentrer sur la charge la plus importante apportée par la requête sélectionnée.


Database load
Current activity measured in average active sessions (AAS)
Bar
Slice by wait
Show max vCPU
Average active sessions (AAS)
12
10
8
6
4
2
0
04:49
04:50
04:51
04:52
04:53
04:54
04:55
04:56
04:57
04:58
04:59
05:00
05:01
05:02
Time (UTC)
CPU
IO
Latch
Other
Max vCPUs
Top waits
Top queries
Top hosts
Top databases
Top applications
Top queries (3)
Learn more
Find query statements
Load by wait (AAS)
Query statements
2.34
{"find":"customerscollection","filter":{"FirstName":"?"},"sort":{"key":{"$number...
2.05
{"find":"usertable","filter":{"_id":"?"},"limit":{"$numberInt":"?"},"singleBatch...
< 0.01
{"ismaster":{"$numberInt":"?"},"helloOk":"?","$db":"admin","$readPreference":{"m...



Vue d'ensemble de l'onglet Requêtes les plus fréquentes

Par défaut, l'onglet Requête principale affiche les requêtes qui contribuent le plus à la charge de la base de données. Vous pouvez analyser le texte de la requête pour affiner vos requêtes.

Rubriques

- Résumés de requêtes

- Load by waits (AAS) [Charge par attentes (AAS)]
- Affichage des informations détaillées sur les requêtes
- Accès au texte d'une requête de déclaration
- Afficher et télécharger le texte d'une requête de relevé

Résumés de requêtes

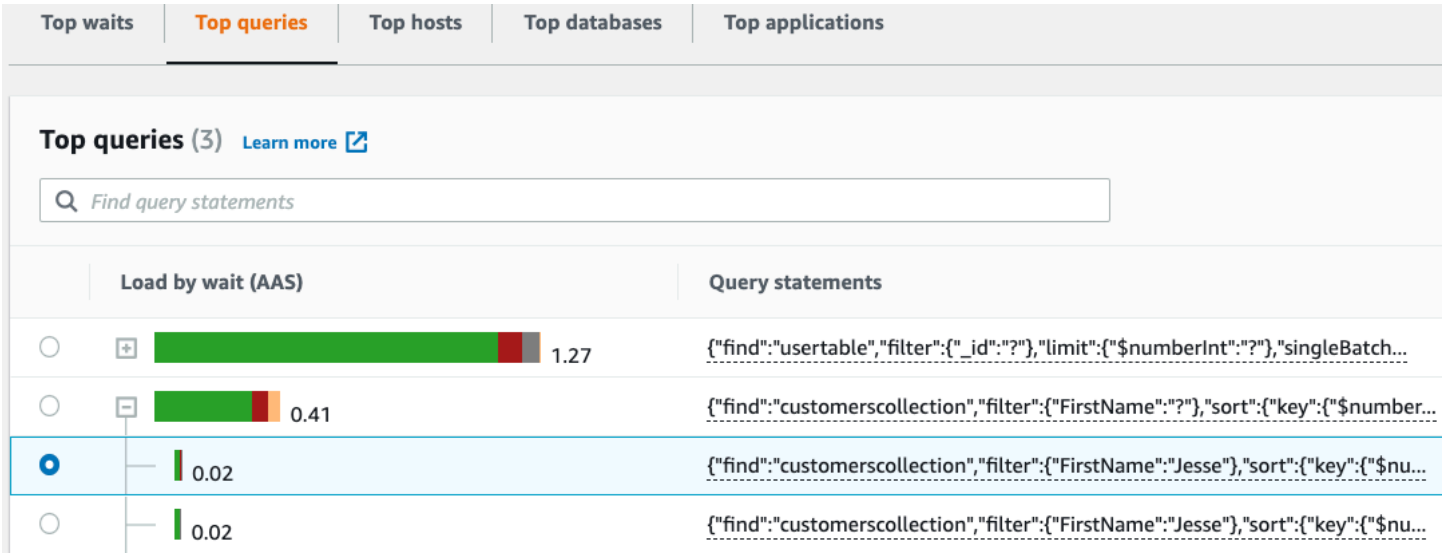
Un résumé de requête est un composite de plusieurs requêtes réelles qui sont structurellement similaires mais qui peuvent avoir des valeurs littérales différentes. Le récapitulatif remplace les valeurs codées en dur par un point d'interrogation. Par exemple, un résumé de requête peut ressembler à ceci :

```
{"find":"customerscollection","filter":{"FirstName":"?"},"sort":{"key":
{"$numberInt":"?"}},"limit":{"$numberInt":"?"}}
```

Ce récapitulatif peut inclure les requêtes enfant suivantes :

```
{"find":"customerscollection","filter":{"FirstName":"Karrie"},"sort":{"key":
{"$numberInt":"1"}},"limit":{"$numberInt":"3"}}
{"find":"customerscollection","filter":{"FirstName":"Met"},"sort":{"key":
{"$numberInt":"1"}},"limit":{"$numberInt":"3"}}
{"find":"customerscollection","filter":{"FirstName":"Rashin"},"sort":{"key":
{"$numberInt":"1"}},"limit":{"$numberInt":"3"}}
```

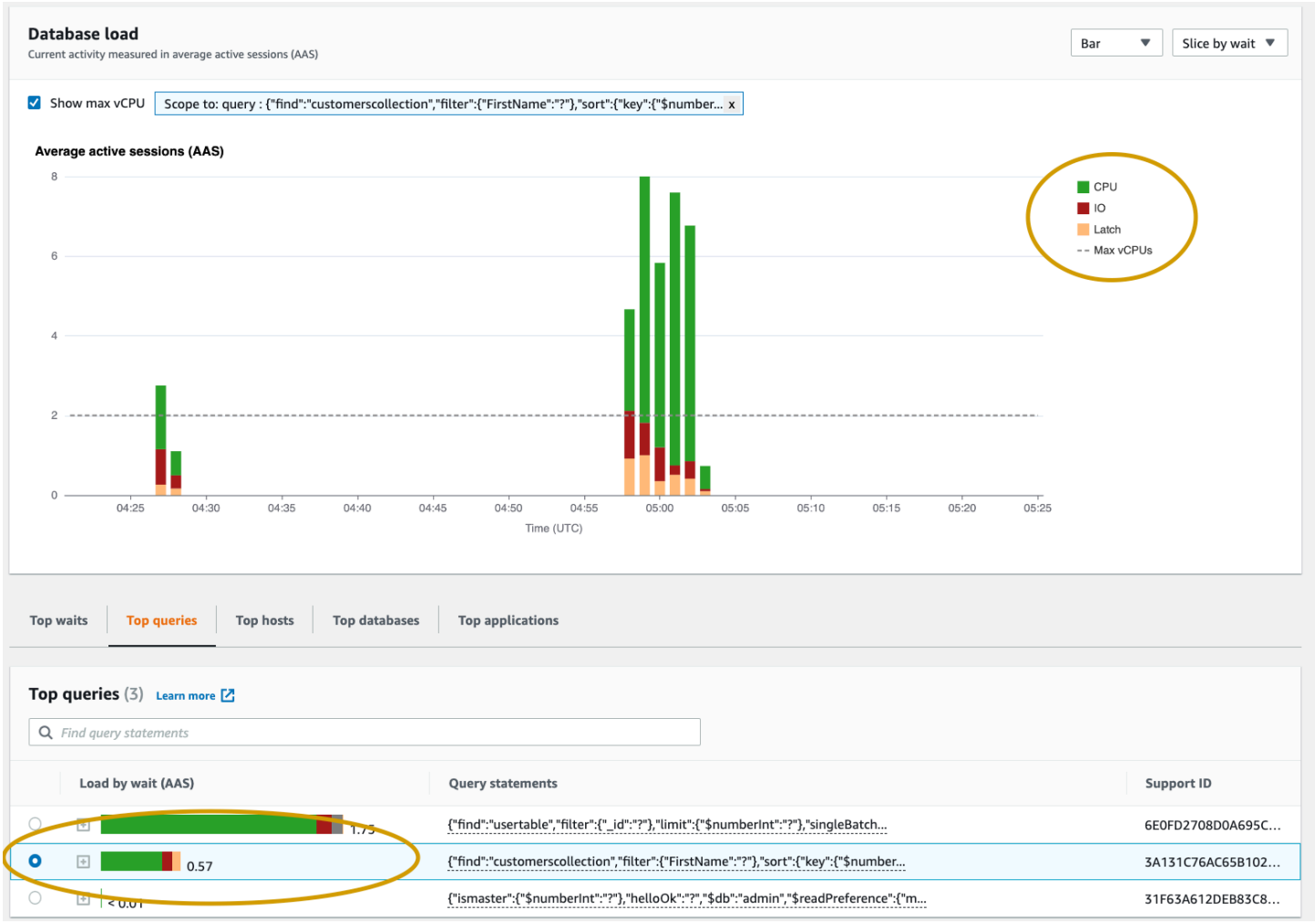
Pour voir les instructions de requête littérales dans un résumé, sélectionnez la requête, puis le symbole plus (+). Dans la capture d'écran suivante, la requête sélectionnée est un récapitulatif.



> **Note**
>
> Un résumé de requête regroupe des instructions de requête similaires, mais ne supprime pas les informations sensibles.

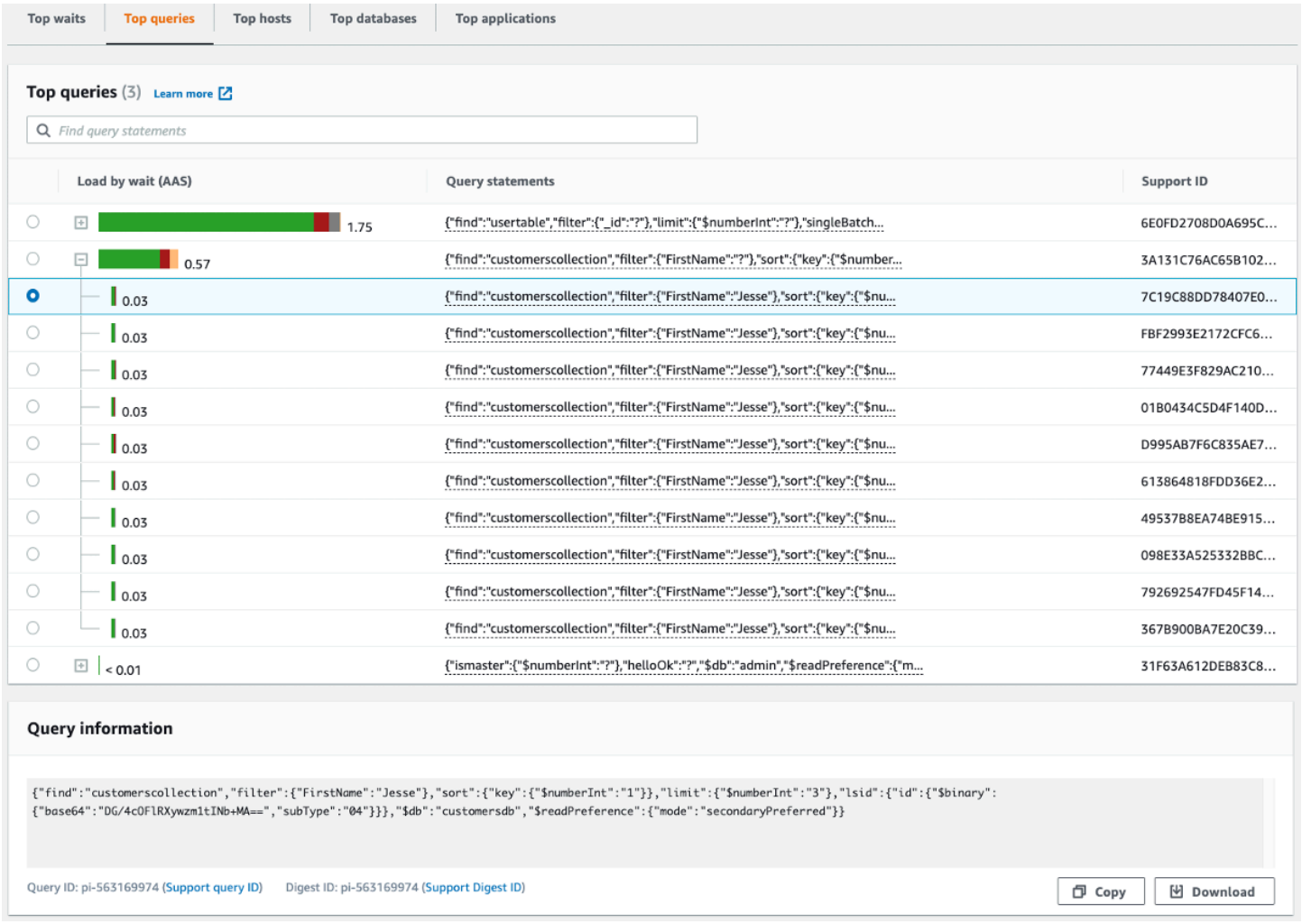
Load by waits (AAS) [Charge par attentes (AAS)]

Dans les requêtes les plus importantes, la colonne Load by waits (AAS) illustre le pourcentage de charge de base de données associé à chaque élément de charge maximale. Cette colonne reflète la charge de cet élément selon le groupe actuellement sélectionné dans le graphique de charge de la base de données. Par exemple, vous pouvez regrouper le graphique DB Load (Charge de la base de données) par états d'attente. Dans ce cas, la dimension, la segmentation et le code de couleurs de la barre DB Load by Waits (Charge de base de données par attente) représentent la proportion du temps d'un état d'attente donné auquel cette requête contribue. Cette barre indique également les états d'attente qui affectent la requête sélectionnée.



Affichage des informations détaillées sur les requêtes

Dans la table des requêtes Top, vous pouvez ouvrir une instruction de synthèse pour afficher ses informations. Les informations s'affichent dans le volet inférieur.



Les types d'identifiants (ID) suivants sont associés aux instructions de requête :

1. ID de requête de support : valeur de hachage de l'ID de requête. Cette valeur sert uniquement à référencer un ID de requête lorsque vous travaillez avec AWS Support. AWS Support n'a pas accès à vos identifiants de requête réels ni au texte de votre requête.
2. Support Digest ID : valeur de hachage de l'ID digest. Cette valeur est uniquement destinée à référencer un ID digest lorsque vous utilisez AWS Support. AWS Support n'a pas accès à vos identifiants de résumé ni au texte de votre requête.

Accès au texte d'une requête de déclaration

Par défaut, chaque ligne du tableau des requêtes les plus fréquentes affiche 500 octets de texte de requête pour chaque instruction de requête. Lorsqu'une instruction de synthèse dépasse 500 octets, vous pouvez afficher plus de texte en ouvrant l'instruction dans le tableau de bord Performance Insights. Dans ce cas, la longueur maximale de la requête affichée est de 1 Ko. Si vous consultez un énoncé de requête complet, vous pouvez également choisir Télécharger.

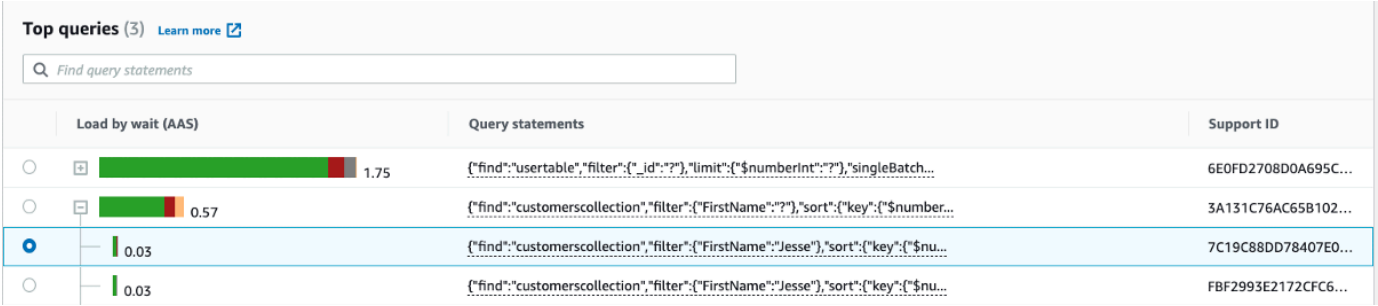
Afficher et télécharger le texte d'une requête de relevé

Dans le tableau de bord Performance Insights, vous pouvez consulter ou télécharger le texte de la requête.

Pour afficher davantage de texte de requête dans le tableau de bord Performance Insights

1. Ouvrez la console Amazon DocumentDB à l'adresse : https://console.aws.amazon.com/docdb/
2. Dans le volet de navigation, choisissez Performance Insights.
3. Choisissez une instance de base de données. Le tableau de bord de Performance Insights s'affiche pour cette instance de base de données.

   Les instructions de requête dont le texte est supérieur à 500 octets ressembleront à l'image suivante :



4. Consultez la section des informations sur la requête pour voir une plus grande partie du texte de la requête.

Le tableau de bord Performance Insights peut afficher jusqu'à 1 Ko pour chaque instruction de requête complète.

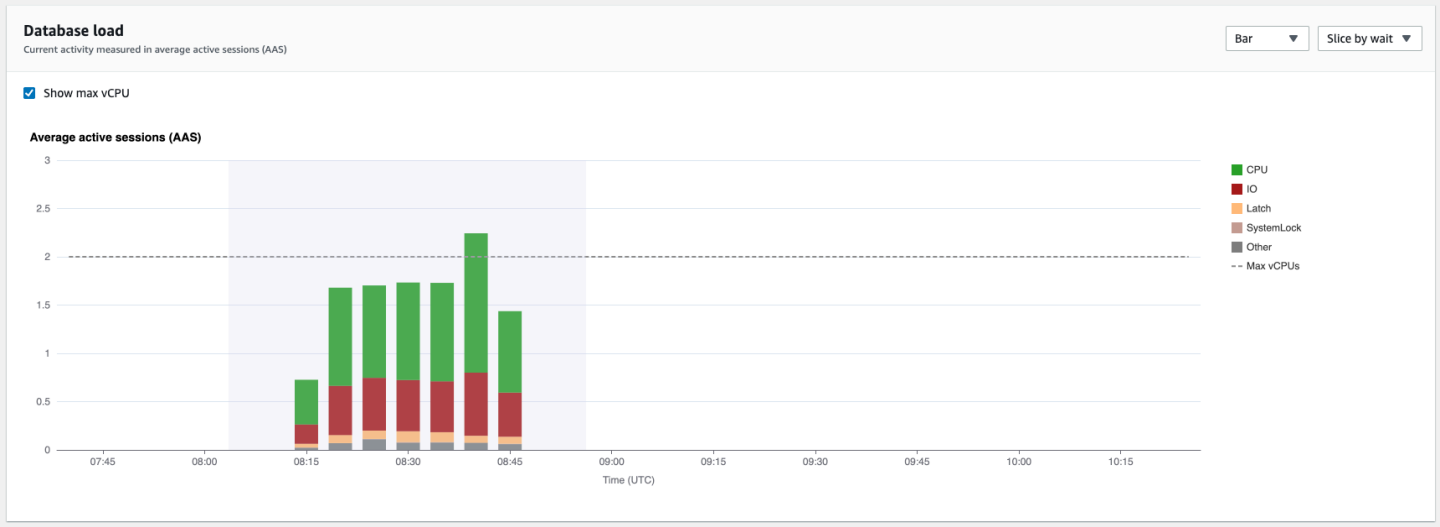
> **Note**
>
> Pour copier ou télécharger l'instruction de requête, désactivez les bloqueurs de fenêtres contextuelles.

Zoomer sur le graphique de charge de la base de données

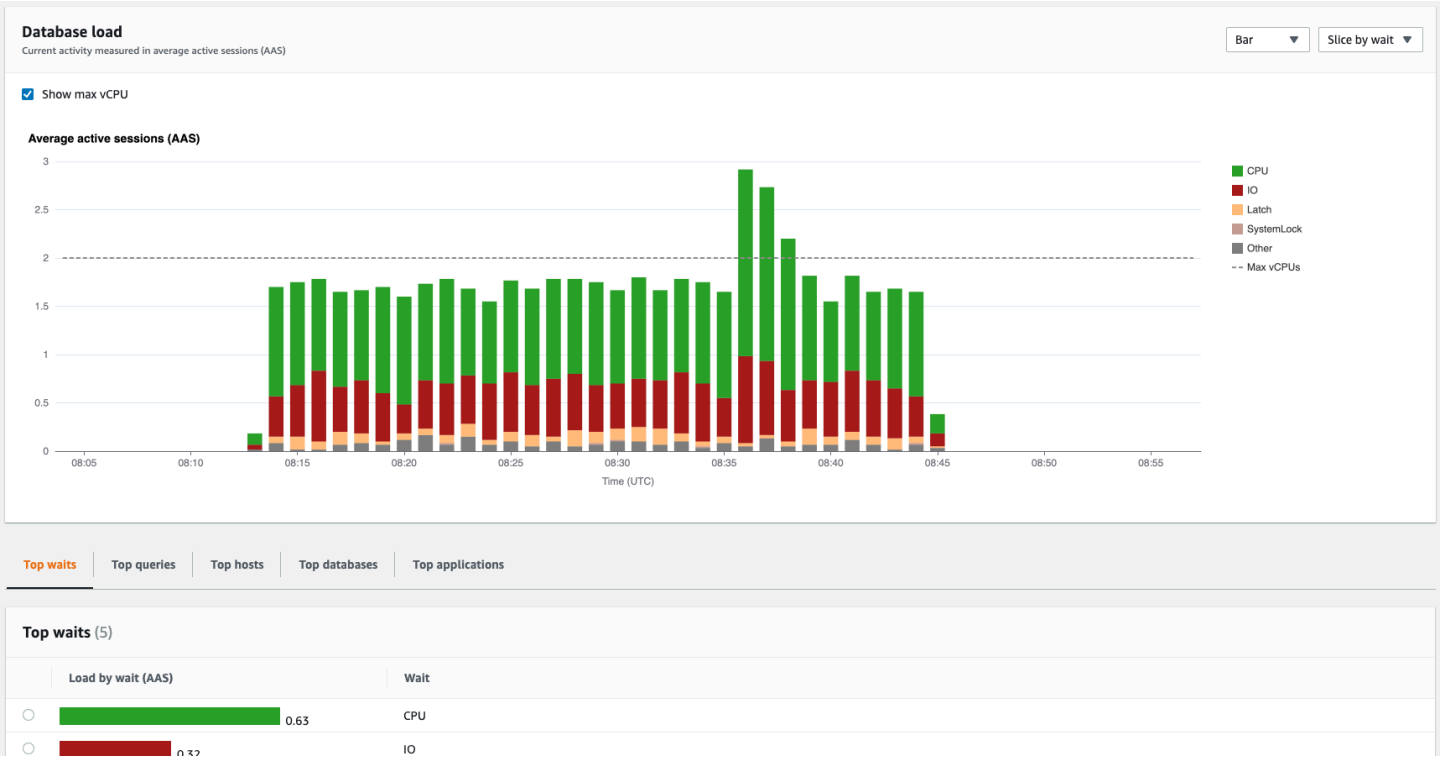
Vous pouvez utiliser d'autres fonctions de l'interface utilisateur Performance Insights pour vous aider à analyser les données de performance.

Zoom avant par cliquer et glisser

Dans l'interface Performance Insights, vous pouvez sélectionner une petite portion du graphique de charge et faire un zoom avant sur les détails.



Faites un zoom avant sur une portion du graphique de charge, choisissez l'heure de début et faites glisser jusqu'à la fin de la période souhaitée. Lorsque vous faites ceci, la zone sélectionnée est mise en évidence. Lorsque vous relâchez la souris, le graphique de charge zoome sur la zone sélectionnée et le tableau des principaux éléments est recalculé.



Récupération de métriques avec l'API Performance Insights

Lorsque l'API Performance Insights est activée, elle fournit une visibilité sur les performances des instances. Amazon CloudWatch Logs fournit la source officielle pour les métriques de surveillance des services vendus. AWS

Performance Insights offre une vue spécifique au domaine de la charge de base de données mesurée en tant que moyenne des sessions actives (AAS). Cette métrique est présentée aux consommateurs de l'API sous la forme d'un ensemble de données de série chronologique bidimensionnel. La dimension temporelle des données fournit les données de charge de la base de données pour chaque point temporel de la plage de temps interrogée. Chaque point dans te temps décompose la charge globale par rapport aux dimensions demandées, par exemple, `Query`, `Wait-state`, `Application` ou `Host`, mesurée à ce point dans le temps.

Amazon DocumentDB Performance Insights surveille votre instance de base de données Amazon DocumentDB afin que vous puissiez analyser et résoudre les problèmes liés aux performances de la base de données. Vous pouvez consulter les données de Performance Insights dans AWS Management Console. Performance Insights fournit également une API publique qui vous permet d'interroger vos propres données. Vous pouvez utiliser l'API pour effectuer les opérations suivantes :

- Déchargement des données dans une base de données

- Ajout de données Performance Insights aux tableaux de bord de surveillance existants
- Création d'outils de surveillance

Pour utiliser l'API Performance Insights, activez Performance Insights sur l'une de vos instances Amazon DocumentDB. Pour de plus amples informations sur l'activation de Performance Insights, veuillez consulter Activation et désactivation de Performance Insights. Pour de plus amples informations sur l'API Performance Insights, veuillez consulter la Référence d'API Performance Insights.

L'API Performance Insights fournit les opérations suivantes.

Action Performance Insights	AWS CLI commande	Description
`DescribeDimensionKeys`	`aws pi describe-dimension-keys`	Récupère les N premières clés de dimension d'une mesure sur une période spécifique.
`GetDimensionKeyDetails`	`aws pi get-dimension-key-details`	Récupère les attributs du groupe de dimensions spécifié pour une instance de base de données ou une source de données. Par exemple, si vous spécifiez un ID de requête et si les détails de la dimension sont disponibles, le texte intégral de la dimension `db.query.statement` associée à cet ID est `GetDimensionKeyDetails` extrait. Cette opération est utile car elle `GetResourceMetrics` `DescribeDimensionKeys` ne permet pas de récupérer un texte d'instruction de requête volumineux.

Action Performance Insights	AWS CLI commande	Description
GetResourceMetadata	aws pi get-resource-metadata	Récupérez les métadonnées de différentes fonctions. Par exemple, les métadonnées peuvent indiquer qu'une fonction est activée ou désactivée sur une instance de base de données spécifique.
GetResourceMetrics	aws pi get-resource-metrics	Récupère les métriques Performance Insights d'un ensemble de sources de données, au cours d'une période. Vous pouvez fournir des groupes de dimensions et des dimensions spécifiques, ainsi que des critères d'agrégation et de filtrage, pour chaque groupe.
ListAvailableResourceDimensions	aws pi list-available-resource-dimensions	Récupérez les dimensions pouvant être interrogées pour chaque type de métrique spécifié sur une instance spécifiée.
ListAvailableResourceMetrics	aws pi list-available-resource-metrics	Récupérez toutes les métriques disponibles des types de métriques spécifiés pouvant être interrogés pour une instance de base de données spécifiée.

Rubriques

- AWS CLI pour Performance Insights
- Récupération de métriques de série chronologique
- AWS CLIExemples pour Performance Insights

AWS CLI pour Performance Insights

Vous pouvez consulter les données de Performance Insights à l'aide d AWS CLI. Vous pouvez obtenir de l'aide sur les commandes AWS CLI relatives à Performance Insights en saisissant le code suivant sur la ligne de commande.

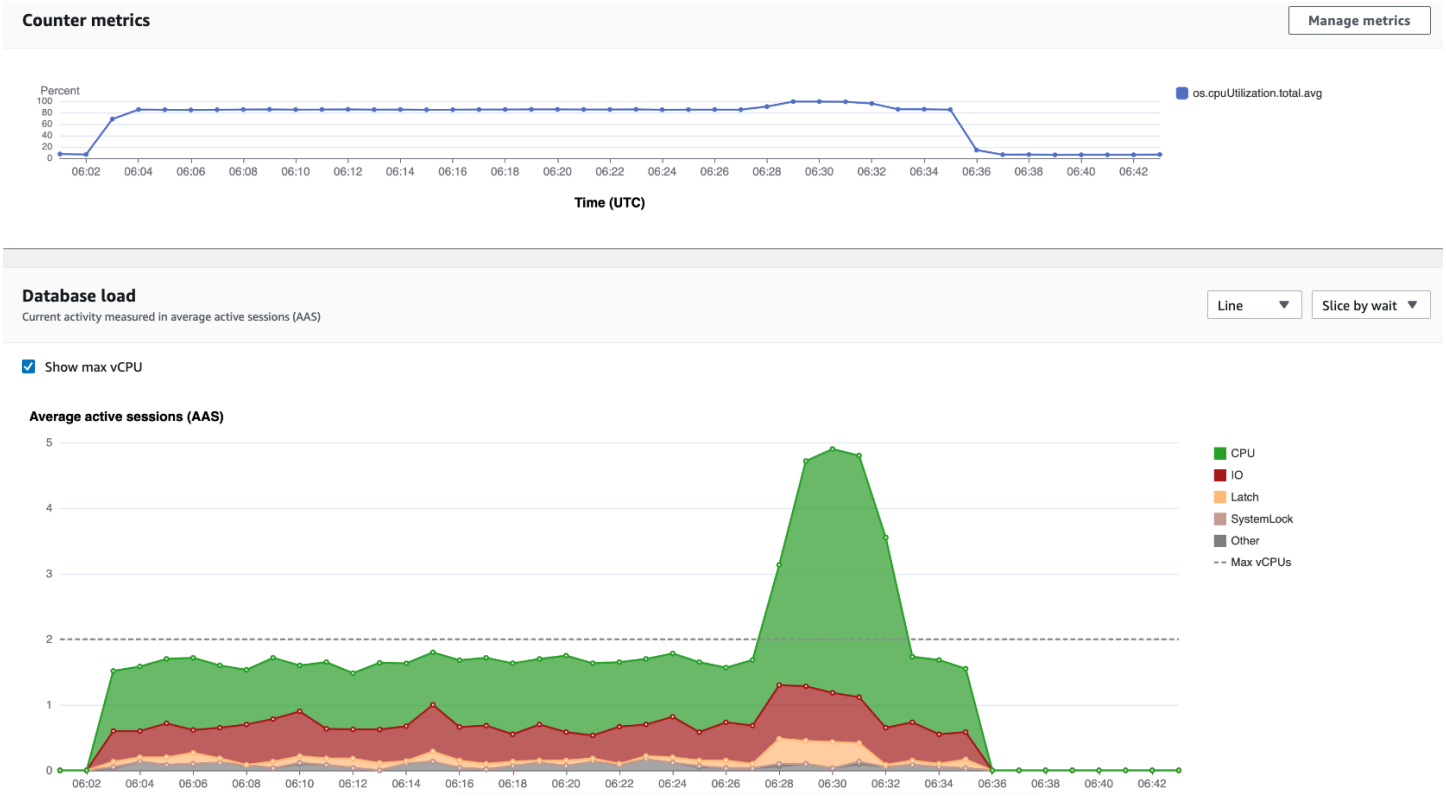
```
aws pi help
```

Si vous n'avez pas installé l'AWS CLI, veuillez consulter la rubrique Installation de l'interface de ligne de commande AWS dans le Guide de l'utilisateur AWS CLI pour en savoir plus sur son installation.

Récupération de métriques de série chronologique

L'opération `GetResourceMetrics` récupère une ou plusieurs métriques de série chronologique à partir des données de Performance Insights. `GetResourceMetrics` exige une métrique et une période, et renvoie une réponse contenant la liste des points de données.

Par exemple, AWS Management Console utilise `GetResourceMetrics` pour renseigner le graphique Counter Metrics (Métriques de compteur) et le graphique Database Load (Charge de base de données), comme illustré dans l'image ci-dessous.



Toutes les métriques renvoyées par `GetResourceMetrics` sont des métriques de série chronologique standard, à l'exception de `db.load`. Elle apparaît dans le graphique Database Load (Charge de base de données). La métrique `db.load` est différente des autres métriques de série chronologique, car vous pouvez la décomposer en sous-composants appelés dimensions. Dans l'image précédente, `db.load` est décomposé et regroupé en fonction des états d'attente qui constituent `db.load`.

> **Note**
>
> `GetResourceMetrics` peut également renvoyer la métrique `db.sampleload`, mais la métrique `db.load` est appropriée dans la plupart des cas.

Pour de plus amples informations sur les métriques de compteur renvoyées par `GetResourceMetrics`, veuillez consulter Performance Insights pour les contre-métriques.

Les calculs suivants sont pris en charge pour les métriques :

- Moyenne – Moyenne de la métrique sur une période. Ajoutez `.avg` au nom de la métrique.
- Minimum – Valeur minimale de la métrique sur une période. Ajoutez `.min` au nom de la métrique.

- Maximum – Valeur maximale de la métrique sur une période. Ajoutez `.max` au nom de la métrique.
- Somme – Somme des valeurs de la métrique sur une période. Ajoutez `.sum` au nom de la métrique.
- Nombre échantillon – Nombre de fois où la métrique a été collectée sur une période. Ajoutez `.sample_count` au nom de la métrique.

Par exemple, supposons qu'une métrique soit collectée pendant 300 secondes (5 minutes) et qu'elle soit collectée une fois toutes les minutes. Les valeurs pour chaque minute sont 1, 2, 3, 4 et 5. Dans ce cas, les calculs suivants sont renvoyés :

- Moyenne – 3
- Minimum – 1
- Maximum – 5
- Somme – 15
- Nombre échantillon – 5

Pour plus d'informations sur l'utilisation de la commande AWS CLI `get-resource-metrics`, consultez `get-resource-metrics`.

Pour l'option `--metric-queries`, spécifiez une ou plusieurs requêtes pour lesquelles vous souhaitez obtenir les résultats. Chaque requête se compose d'un paramètre `Metric` obligatoire et des paramètres `GroupBy` et `Filter` facultatifs. Voici un exemple de spécification de l'option `--metric-queries`.

```
{
   "Metric": "string",
   "GroupBy": {
     "Group": "string",
     "Dimensions": ["string", ...],
     "Limit": integer
   },
   "Filter": {"string": "string"
     ...}
```

AWS CLIExemples pour Performance Insights

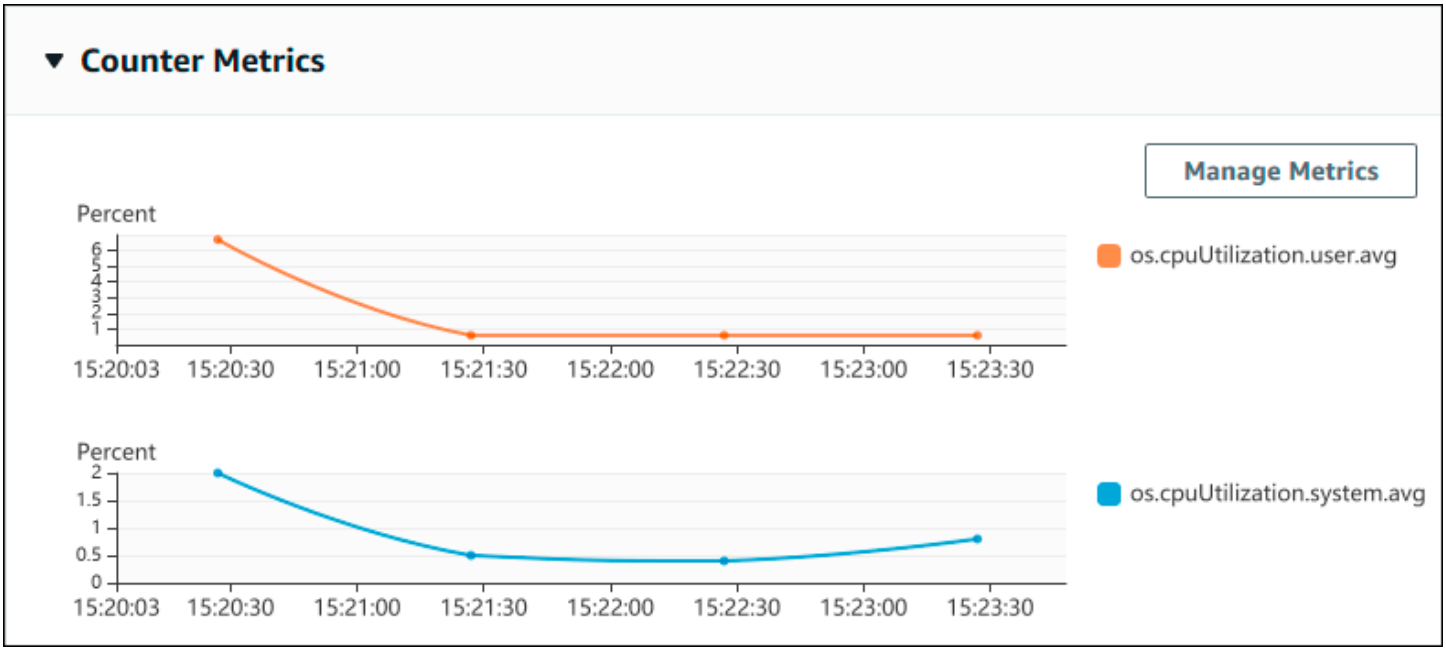
Les exemples suivants montrent comment utiliser l'AWS CLI pour Performance Insights.

Rubriques

- Récupération de métriques de compteur
- Récupération de la charge moyenne de la base de données pour les états d'attente les plus élevés
- Récupération de la charge moyenne de la base de données pour la requête la plus élevée
- Récupération de la charge moyenne de la base de données filtrée par requête

Récupération de métriques de compteur

L'image suivante illustre deux graphiques de métriques de compteur dans AWS Management Console.



L'exemple suivant décrit comment collecter les mêmes données utilisées par AWS Management Console pour générer les deux graphiques Counter Metrics (Métriques de compteur).

Pour Linux, macOS ou Unix :

```
aws pi get-resource-metrics \
   --service-type DOCDB \
   --identifier db-ID \
   --start-time 2022-03-13T8:00:00Z \
   --end-time   2022-03-13T9:00:00Z \
   --period-in-seconds 60 \
   --metric-queries '[{"Metric": "os.cpuUtilization.user.avg"  },
                      {"Metric": "os.cpuUtilization.idle.avg"}]'
```

Pour Windows :

```
aws pi get-resource-metrics ^
   --service-type DOCDB ^
   --identifier db-ID ^
   --start-time 2022-03-13T8:00:00Z ^
   --end-time   2022-03-13T9:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries '[{"Metric": "os.cpuUtilization.user.avg"  },
                      {"Metric": "os.cpuUtilization.idle.avg"}]'
```

Vous pouvez également simplifier la lecture d'une commande en spécifiant un fichier pour l'option `--metrics-query`. L'exemple suivant utilise un fichier nommé query.json pour l'option. Le contenu du fichier est le suivant.

```
[
    {
        "Metric": "os.cpuUtilization.user.avg"
    },
    {
        "Metric": "os.cpuUtilization.idle.avg"
    }
]
```

Exécutez la commande suivante pour utiliser le fichier.

Pour Linux, macOS ou Unix :

```
aws pi get-resource-metrics \
   --service-type DOCDB \
   --identifier db-ID \
   --start-time 2022-03-13T8:00:00Z \
   --end-time   2022-03-13T9:00:00Z \
   --period-in-seconds 60 \
   --metric-queries file://query.json
```

Pour Windows :

```
aws pi get-resource-metrics ^
   --service-type DOCDB ^
   --identifier db-ID ^
   --start-time 2022-03-13T8:00:00Z ^
```

```
    --end-time   2022-03-13T9:00:00Z ^
    --period-in-seconds 60 ^
    --metric-queries file://query.json
```

L'exemple précédent spécifie les valeurs suivantes pour les options :

- `--service-type`— DOCDB pour Amazon DocumentDB
- `--identifier` – ID de ressource de l'instance de base de données
- `--start-time` et `--end-time` – Valeurs `DateTime` conformes à l'ISO 8601 pour la période à interroger, avec plusieurs formats pris en charge

L'interrogation se déroule pendant un intervalle d'une heure :

- `--period-in-seconds` – `60` pour une requête toutes les minutes
- `--metric-queries` – Tableau de deux requêtes s'appliquant chacune à une métrique.

  Le nom de la métrique utilise des points pour classifier la métrique dans une catégorie utile, l'élément final étant une fonction. Dans l'exemple, la fonction est `avg` pour chaque requête. Comme pour Amazon CloudWatch, les fonctions prises en charge sont `min`,`max`,`total`, et`avg`.

La réponse ressemble à ce qui suit.

```
{
    "AlignedStartTime": "2022-03-13T08:00:00+00:00",
    "AlignedEndTime": "2022-03-13T09:00:00+00:00",
    "Identifier": "db-NQF3TTMFQ3GTOKIMJODMC3KQQ4",
    "MetricList": [
        {
            "Key": {
                "Metric": "os.cpuUtilization.user.avg"
            },
            "DataPoints": [
                {
                    "Timestamp": "2022-03-13T08:01:00+00:00", //Minute1
                    "Value": 3.6
                },
                {
                    "Timestamp": "2022-03-13T08:02:00+00:00", //Minute2
                    "Value": 2.6
                },
```

```
                //.... 60 datapoints for the os.cpuUtilization.user.avg metric
        {
            "Key": {
                "Metric": "os.cpuUtilization.idle.avg"
            },
            "DataPoints": [
                {
                    "Timestamp": "2022-03-13T08:01:00+00:00",
                    "Value": 92.7
                },
                {
                    "Timestamp": "2022-03-13T08:02:00+00:00",
                    "Value": 93.7
                },
                //.... 60 datapoints for the os.cpuUtilization.user.avg metric
            ]
        }
    ] //end of MetricList
} //end of response
```

La réponse contient les éléments `Identifier`, `AlignedStartTime` et `AlignedEndTime`. Étant donné que la valeur de `--period-in-seconds` était définie sur `60`, les heures de début et de fin ont été arrondies à la minute près. Si `--period-in-seconds` était défini sur `3600`, les heures de début et de fin auraient été arrondies à l'heure près.

L'élément `MetricList` dans la réponse comporte un certain nombre d'entrées, chacune associée à une entrée `Key` et `DataPoints`. Chaque élément `DataPoint` comporte une entrée `Timestamp` et `Value`. Chaque liste `Datapoints` répertorie 60 points de données, car les requêtes sont exécutées toutes les minutes pendant une heure, avec `Timestamp1/Minute1`, `Timestamp2/Minute2`, etc. jusqu'à `Timestamp60/Minute60`.

Étant donné que la requête s'applique à deux métriques de compteur différentes, contient deux élément `MetricList`.

Récupération de la charge moyenne de la base de données pour les états d'attente les plus élevés

L'exemple suivant illustre la même requête utilisée par AWS Management Console pour générer un graphique en aires empilées. Cet exemple extrait le résultat de la `db.load.avg` dernière heure en divisant la charge en fonction des sept premiers états d'attente. La commande est identique à la commande de la rubrique Récupération de métriques de compteur. Le contenu du fichier query.json est cependant différent :

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_state", "Limit": 7 }
    }
]
```

Exécutez la commande suivante.

Pour Linux, macOS ou Unix :

```
aws pi get-resource-metrics \
   --service-type DOCDB \
   --identifier db-ID \
   --start-time 2022-03-13T8:00:00Z \
   --end-time   2022-03-13T9:00:00Z \
   --period-in-seconds 60 \
   --metric-queries file://query.json
```

Pour Windows :

```
aws pi get-resource-metrics ^
   --service-type DOCDB ^
   --identifier db-ID ^
   --start-time 2022-03-13T8:00:00Z ^
   --end-time   2022-03-13T9:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries file://query.json
```

L'exemple indique la métrique de `db.load.avg` et a `GroupBy` des sept premiers états d'attente. Pour plus de détails sur les valeurs valides pour cet exemple, consultez DimensionGrouple manuel Performance Insights API Reference.

La réponse ressemble à ce qui suit.

```
{
    "AlignedStartTime": "2022-04-04T06:00:00+00:00",
    "AlignedEndTime": "2022-04-04T06:15:00+00:00",
    "Identifier": "db-NQF3TTMFQ3GTOKIMJODMC3KQQ4",
    "MetricList": [
        {//A list of key/datapoints
            "Key": {
```

```
                //A Metric with no dimensions. This is the total db.load.avg
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                //Each list of datapoints has the same timestamps and same number of
 items
                {
                    "Timestamp": "2022-04-04T06:01:00+00:00",//Minute1
                    "Value": 0.0
                },
                {
                    "Timestamp": "2022-04-04T06:02:00+00:00",//Minute2
                    "Value": 0.0
                },
                //... 60 datapoints for the total db.load.avg key
                ]
        },
        {
            "Key": {
                //Another key. This is db.load.avg broken down by CPU
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_state.name": "CPU"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": "2022-04-04T06:01:00+00:00",//Minute1
                    "Value": 0.0
                },
                {
                    "Timestamp": "2022-04-04T06:02:00+00:00",//Minute2
                    "Value": 0.0
                },
                //... 60 datapoints for the CPU key
            ]
        },//... In total we have 3 key/datapoints entries, 1) total, 2-3) Top Wait
 States
    ] //end of MetricList
} //end of response
```

Dans cette réponse, il y a trois entrées dans le`MetricList`. Il y a une entrée pour le totaldb.load.avg, et trois entrées chacune pour le résultat db.load.avg divisé en fonction de l'un des trois premiers états d'attente. Comme il existait une dimension de regroupement (contrairement au premier exemple), il doit y avoir une clé pour chaque regroupement de la métrique. Un seul élément Key peut être associé à chaque métrique, comme dans le cas d'utilisation de la métrique de compteur de base.

Récupération de la charge moyenne de la base de données pour la requête la plus élevée

L'exemple suivant regroupe db.wait_state les 10 principales instructions de requête. Il existe deux groupes différents pour les instructions de requête :

- db.query— L'instruction de requête complète, telle que {"find":"customers","filter": {"FirstName":"Jesse"},"sort":{"key":{"$numberInt":"1"}}}
- db.query_tokenized— L'instruction de requête tokenisée, telle que {"find":"customers","filter":{"FirstName":"?"},"sort":{"key": {"$numberInt":"?"}},"limit":{"$numberInt":"?"}}

Lors de l'analyse des performances d'une base de données, il peut être utile de considérer les instructions de requête dont les paramètres ne diffèrent que comme un élément logique. Vous pouvez donc utiliser db.query_tokenized lors de l'interrogation. Toutefois, en particulier lorsque cela vous intéresseexplain(), il est parfois plus utile d'examiner des instructions de requête complètes avec des paramètres. Il existe une relation parent-enfant entre les requêtes tokenisées et les requêtes complètes, plusieurs requêtes complètes (enfants) étant regroupées sous la même requête tokenisée (parent).

La commande illustrée dans cet exemple est identique à la commande de la rubrique Récupération de la charge moyenne de la base de données pour les états d'attente les plus élevés. Le contenu du fichier query.json est cependant différent :

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.query_tokenized", "Limit": 10 }
    }
]
```

L'exemple suivant utilise db.query_tokenized.

Pour Linux, macOS ou Unix :

```
aws pi get-resource-metrics \
   --service-type DOCDB \
   --identifier db-ID \
   --start-time 2022-03-13T8:00:00Z \
   --end-time   2022-03-13T9:00:00Z \
   --period-in-seconds 3600 \
   --metric-queries file://query.json
```

Pour Windows :

```
aws pi get-resource-metrics ^
   --service-type DOCDB ^
   --identifier db-ID ^
   --start-time 2022-03-13T8:00:00Z ^
   --end-time   2022-03-13T9:00:00Z  ^
   --period-in-seconds 3600 ^
   --metric-queries file://query.json
```

Cet exemple montre des requêtes de plus d'une heure, avec une minute period-in-seconds.

L'exemple indique la métrique de `db.load.avg` et a `GroupBy` des sept premiers états d'attente. Pour plus de détails sur les valeurs valides pour cet exemple, consultez DimensionGrouple manuel Performance Insights API Reference.

La réponse ressemble à ce qui suit.

```
{
    "AlignedStartTime": "2022-04-04T06:00:00+00:00",
    "AlignedEndTime": "2022-04-04T06:15:00+00:00",
    "Identifier": "db-NQF3TTMFQ3GTOKIMJODMC3KQQ4",
    "MetricList": [
        {//A list of key/datapoints
            "Key": {
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                //... 60 datapoints for the total db.load.avg key
                ]
        },
```

```
               {
            "Key": {//Next key are the top tokenized queries
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.query_tokenized.db_id": "pi-1064184600",
                    "db.query_tokenized.id": "77DE8364594EXAMPLE",
                    "db.query_tokenized.statement": "{\"find\":\"customers\",\"filter
\":{\"FirstName\":\"?\"},\"sort\":{\"key\":{\"$numberInt\":\"?\"}},\"limit\"
:{\"$numberInt\":\"?\"},\"$db\":\"myDB\",\"$readPreference\":{\"mode\":\"primary\"}}"
                }
            },
            "DataPoints": [
            //... 60 datapoints
            ]
        },
        // In total 11 entries, 10 Keys of top tokenized queries, 1 total key
    ] //End of MetricList
} //End of response
```

Cette réponse comporte 11 entrées `MetricList` (1 au total, 10 requêtes les plus tokenisées), chaque entrée en comptant 24 par heure. `DataPoints`

Pour les requêtes tokenisées, chaque liste de dimensions comporte trois entrées :

- `db.query_tokenized.statement`— L'instruction de requête tokenisée.
- `db.query_tokenized.db_id` — L'identifiant synthétique que Performance Insights génère pour vous. Cet exemple renvoie l'ID synthétique `pi-1064184600`.
- `db.query_tokenized.id` – ID de la requête dans Performance Insights.

  Dans AWS Management Console, cet ID se nomme ID de support. Il porte ce nom, car l'ID représente les données qu'AWS Support peut examiner pour vous aider à résoudre un problème lié à votre base de données. AWS prend la sécurité et la confidentialité de vos données très au sérieux, et presque toutes les données sont stockées en mode chiffré avec votre clé principale client (CMK) AWS KMS. Personne au sein d'AWS ne peut ainsi consulter ces données. Dans l'exemple précédent, `tokenized.statement` et `tokenized.db_id` sont tous les deux stockés sous forme chiffrée. Si vous rencontrez un problème avec votre base de données, AWS Support peut vous aider en fournissant l'ID de support.

Lors de l'interrogation, il peut s'avérer utile de spécifier une entrée `Group` dans `GroupBy`. Toutefois, pour contrôler les données renvoyées de manière plus précise, spécifier la liste des dimensions.

Par exemple, si `db.query_tokenized.statement` est le seul élément nécessaire, un attribut `Dimensions` peut être ajouté au fichier query.json.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": {
            "Group": "db.query_tokenized",
            "Dimensions":["db.query_tokenized.statement"],
            "Limit": 10
        }
    }
]
```

Récupération de la charge moyenne de la base de données filtrée par requête

La requête d'API correspondante illustrée dans cet exemple est identique à la commande de la rubrique Récupération de la charge moyenne de la base de données pour la requête la plus élevée. Le contenu du fichier query.json est cependant différent :

```
[
 {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_state", "Limit": 5  },
        "Filter": { "db.query_tokenized.id": "AKIAIOSFODNN7EXAMPLE" }
    }
]
```

Dans cette réponse, toutes les valeurs sont filtrées en fonction de la contribution de la requête tokenisée AKIAIOSFODNN7EXAMPLE spécifiée dans le fichier query.json. Les clés peuvent également suivre un ordre différent de celui d'une requête sans filtre, car ce sont les cinq premiers états d'attente qui ont affecté la requête filtrée.

CloudWatch Métriques Amazon pour Performance Insights

Performance Insights publie automatiquement les statistiques sur Amazon CloudWatch. Les mêmes données peuvent être consultées à partir de Performance Insights, mais l'ajout des métriques CloudWatch facilite l'ajout CloudWatch d'alarmes. ainsi que l'ajout des métriques à des tableaux de bord CloudWatch existants.

Métrique	Description
DBLoad	Le nombre de sessions actives pour Amazon DocumentDB. Vous souhaitez généralem ent obtenir les données relatives au nombre moyen de sessions actives. Dans Performance Insights, ces données sont interrogées sous la forme `db.load.avg` .
DBLoadCPU	Le nombre de sessions actives où le type d'état d'attente est CPU. Dans Performance Insights, ces données sont demandées et filtrées en fonction `db.load.avg` du type d'état d'attente. `CPU`
LoadNonCPU DB	Le nombre de sessions actives où le type d'état d'attente n'est pas CPU.

> **Note**
>
> Ces métriques ne sont publiées CloudWatch que si l'instance de base de données est chargée.

Vous pouvez examiner ces métriques à l'aide de la CloudWatch console, de ou de l' CloudWatchAPI. AWS CLI

Par exemple, vous pouvez obtenir les statistiques de la `DBLoad` métrique en exécutant la get-metric-statisticscommande.

```
aws cloudwatch get-metric-statistics \
    --region ap-south-1 \
    --namespace AWS/DocDB \
    --metric-name DBLoad  \
    --period 360 \
    --statistics Average \
    --start-time 2022-03-14T8:00:00Z \
    --end-time 2022-03-14T9:00:00Z \
```

```
    --dimensions Name=DBInstanceIdentifier,Value=documentdbinstance
```

Cet exemple génère une sortie similaire à la suivante.

```
{
    "Datapoints": [
        {
            "Timestamp": "2022-03-14T08:42:00Z",
            "Average": 1.0,
            "Unit": "None"
        },
        {
            "Timestamp": "2022-03-14T08:24:00Z",
            "Average": 2.0,
            "Unit": "None"
        },
        {
            "Timestamp": "2022-03-14T08:54:00Z",
            "Average": 6.0,
            "Unit": "None"
        },
        {
            "Timestamp": "2022-03-14T08:36:00Z",
            "Average": 5.7,
            "Unit": "None"
        },
        {
            "Timestamp": "2022-03-14T08:06:00Z",
            "Average": 4.0,
            "Unit": "None"
        },
        {
            "Timestamp": "2022-03-14T08:00:00Z",
            "Average": 5.2,
            "Unit": "None"
        }
    ],
    "Label": "DBLoad"
}
```

Vous pouvez utiliser la fonction mathématique des `DB_PERF_INSIGHTS` métriques de la CloudWatch console pour interroger les métriques des compteurs Amazon DocumentDB Performance Insights. La

`DB_PERF_INSIGHTS` fonction inclut également la `DBLoad` métrique à des intervalles inférieurs à la minute. Vous pouvez définir des CloudWatch alarmes sur ces métriques. Pour en savoir plus sur la création d'une alarme, consultez Création d'une alarme sur les métriques de compteur Performance Insights à partir d'une base de données AWS.

Pour plus d'informations CloudWatch, consultez Qu'est-ce qu'Amazon CloudWatch ? dans le guide de CloudWatch l'utilisateur Amazon.

Performance Insights pour les contre-métriques

Les contre-métriques sont des mesures du système d'exploitation figurant dans le tableau de bord Performance Insights. Vous pouvez établir des corrélations entre ces informations et la charge de la base de données pour identifier et analyser les problèmes de performances.

Compteurs de système d'exploitation Performance Insights

Les compteurs de système d'exploitation suivants sont disponibles avec DocumentDB Performance Insights.

Compteur	Type	Métrique
actif	memory	os.memory.active
buffers	memory	os.memory.buffers
mis en cache	memory	os.memory.cached
dirty	memory	os.memory.dirty
free	memory	os.memory.free
inactive	memory	os.memory.inactive
mapped	memory	os.memory.mapped
pageTables	memory	os.memory.pageTables
slab	memory	os.memory.slab
total	memory	os.memory.total

Compteur	Type	Métrique
writeback	memory	os.memory.writeback
idle	cpuUtilization	os.cpuUtilization.idle
system	cpuUtilization	os.cpuUtilization.system
total	cpuUtilization	os.cpuUtilization.total
user	cpuUtilization	os.cpuUtilization.user
wait	cpuUtilization	os.cpuUtilization.wait
one	loadAverageMinute	os. loadAverageMinute.un
fifteen	loadAverageMinute	os. loadAverageMinute.quinze
five	loadAverageMinute	os. loadAverageMinute.cinq
mis en cache	swap	os.swap.cached
free	swap	os.swap.free
dans	swap	os.swap.in
out	swap	os.swap.out
total	swap	os.swap.total
rx	network	os.network.rx
tx	network	os.network.tx
numVCPUs	general	os.general.numVCPUs

Intégration zéro ETL avec Amazon Service OpenSearch

Rubriques

Amazon OpenSearch Service en tant que destination

OpenSearch L'intégration des services à Amazon DocumentDB vous permet de diffuser des événements de chargement complet et de modification des données vers OpenSearch des domaines. L'infrastructure d'ingestion est hébergée sous forme de pipelines d' OpenSearch ingestion et fournit un mécanisme à grande échelle et à faible latence pour diffuser en continu les données des collections Amazon DocumentDB.

Pendant le chargement complet, l'intégration Zero-ETL extrait d'abord les données historiques du chargement complet à OpenSearch l'aide d'un pipeline d'ingestion. Une fois les données à chargement complet ingérées, les pipelines d' OpenSearch ingestion commenceront à lire les données des flux de modifications d'Amazon DocumentDB et finiront par rattraper leur retard afin de maintenir la cohérence des données en temps quasi réel entre Amazon DocumentDB et. OpenSearch OpenSearch stocke les documents dans des index. Les données entrantes provenant d'une collection Amazon DocumentDB peuvent être envoyées vers un index ou peuvent être partitionnées en différents index. Les pipelines d'ingestion synchroniseront tous les événements de création, de mise à jour et de suppression d'une collection Amazon DocumentDB en tant que création, mise à jour et suppression de OpenSearch documents correspondants afin de maintenir la synchronisation des deux systèmes de données. Les pipelines d'ingestion peuvent être configurés pour lire les données d'une collection et écrire dans un index ou lire les données d'une collection et les acheminer de manière conditionnelle vers plusieurs index.

Les pipelines d'ingestion peuvent être configurés pour diffuser des données d'Amazon DocumentDB vers Amazon OpenSearch Service en utilisant :

- Chargement complet uniquement
- Diffusez des événements de flux de modification depuis Amazon DocumentDB sans chargement complet
- Chargement complet suivi de flux de modifications depuis Amazon DocumentDB

Pour configurer votre pipeline d'ingestion, effectuez les étapes suivantes :

Étape 1 : créer un domaine Amazon OpenSearch Service ou une collection OpenSearch sans serveur

Une collection Amazon OpenSearch Service avec les autorisations appropriées pour lire les données est requise. Reportez-vous à la section Getting started with Amazon OpenSearch Service ou Getting started with Amazon OpenSearch Serverless du manuel Amazon OpenSearch Service Developer Guide pour créer une collection. Reportez-vous à OpenSearch Amazon Ingestion dans le manuel Amazon OpenSearch Service Developer Guide pour créer un rôle AIM doté des autorisations appropriées pour accéder aux données d'écriture de la collection ou du domaine.

Étape 2 : activer les flux de modifications sur le cluster Amazon DocumentDB

Assurez-vous que les flux de modifications sont activés sur les collections requises dans le cluster Amazon DocumentDB. Pour plus d'informations, consultez la section Utilisation de Change Streams avec Amazon DocumentDB.

Étape 3 : configurer le rôle de pipeline avec les autorisations d'écriture dans le compartiment Amazon S3 et le domaine ou la collection de destination

Après avoir créé votre collection Amazon DocumentDB et activé le flux de modifications, configurez le rôle de pipeline que vous souhaitez utiliser dans la configuration de votre pipeline et ajoutez-y les autorisations suivantes :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "allowReadAndWriteToS3ForExport",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
```

```
                "arn:aws:s3:::my-bucket/export/*"
            ]
        }
    ]
}
```

Pour qu'un OpenSearch pipeline puisse écrire des données dans un OpenSearch domaine, celui-ci doit disposer d'une politique d'accès au niveau du domaine qui autorise le rôle de pipeline sts_role_arn à y accéder. L'exemple de politique d'accès au domaine suivant permet au rôle de pipeline nommépipeline-role, que vous avez créé à l'étape précédente, d'écrire des données dans le domaine nommé ingestion-domain :

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{your-account-id}:role/{pipeline-role}"
      },
      "Action": ["es:DescribeDomain", "es:ESHttp*"],
      "Resource": "arn:aws:es:{region}:{your-account-id}:domain/{domain-name}/*"
    }
  ]
}
```

Étape 4 : ajouter les autorisations requises sur le rôle de pipeline pour créer X-ENI

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DetachNetworkInterface",
```

```
                "ec2:DescribeNetworkInterfaces"
            ],
            "Resource": [
                "arn:aws:ec2:*:420497401461:network-interface/*",
                "arn:aws:ec2:*:420497401461:subnet/*",
                "arn:aws:ec2:*:420497401461:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [ "ec2:CreateTags" ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
               "StringEquals": { "aws:RequestTag/OSISManaged": "true" }
            }
        }
    ]
}
```

Étape 5 : Création du pipeline

Configurez un pipeline d' OpenSearch ingestion en spécifiant Amazon DocumentDB comme source. Cet exemple de configuration de pipeline suppose l'utilisation d'un mécanisme de récupération des flux de modifications. Reportez-vous à la section Utilisation d'un pipeline d' OpenSearch ingestion avec Amazon DocumentDB dans le manuel Amazon OpenSearch Service Developer Guide pour plus d'informations.

Limites

Les limites suivantes s'appliquent à l'intégration Amazon DocumentDB : OpenSearch

- Une seule collection Amazon DocumentDB comme source par pipeline est prise en charge.
- L'ingestion de données entre régions n'est pas prise en charge. Votre cluster et votre OpenSearch domaine Amazon DocumentDB doivent se trouver dans la même AWS région.
- L'ingestion de données entre comptes n'est pas prise en charge. Votre cluster Amazon DocumentDB et votre pipeline OpenSearch d'ingestion doivent se trouver dans le même AWS compte.
- Les clusters élastiques Amazon DocumentDB ne sont pas pris en charge. Seuls les clusters basés sur des instances Amazon DocumentDB sont pris en charge.
- Assurez-vous que l'authentification est activée à l'aide AWS de secrets sur le cluster Amazon DocumentDB. AWS les secrets sont le seul mécanisme d'authentification pris en charge.
- La configuration du pipeline existante ne peut pas être mise à jour pour ingérer des données provenant d'une autre base de données et/ou d'une autre collection. Pour mettre à jour le nom de base de données et/ou de collection d'un pipeline, vous devez créer un nouveau pipeline.

Développement avec Amazon DocumentDB

Ces sections traitent du développement à l'aide d'Amazon DocumentDB (avec compatibilité avec MongoDB).

Rubriques

Connexion par programmation à Amazon DocumentDB

Cette section contient des exemples de code qui montrent comment se connecter à Amazon DocumentDB (compatible avec MongoDB) en utilisant plusieurs langages différents. Les exemples sont séparés en deux sections selon que vous choisissez de vous connecter à un cluster ayant le protocole TLS (Transport Layer Security) activé ou désactivé. Par défaut, le protocole TLS est activé sur les clusters Amazon DocumentDB. Cependant, vous pouvez désactiver TLS si vous le souhaitez. Pour plus d’informations, consultez Chiffrement des données en transit.

Si vous tentez de vous connecter à votre Amazon DocumentDB depuis l'extérieur du VPC dans lequel réside votre cluster, veuillez consulter Connexion à un cluster Amazon DocumentDB depuis l'extérieur d'un Amazon VPC.

Avant de vous connecter à votre cluster, vous devez savoir si TLS est activé sur celui-ci. La section suivante vous montre comment déterminer la valeur du paramètre `tls` de votre cluster à l'aide de la AWS Management Console ou de l' AWS CLI. Ensuite, vous pouvez continuer en recherchant et en appliquant l'exemple de code approprié.

Rubriques

Déterminer la valeur de votre paramètre `tls`

Déterminer si le protocole TLS est activé sur votre cluster est un processus en deux étapes que vous pouvez effectuer à l'aide du AWS Management Console ou. AWS CLI

1. Déterminez quel groupe de paramètres régit votre cluster.

   Using the AWS Management Console

   1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
   2. Dans le panneau de navigation de gauche, choisissez Clusters.
   3. Dans la liste des clusters, sélectionnez le nom de votre cluster.
   4. La page qui s'affiche répertorie les détails du cluster sélectionné. Faites défiler vers le bas jusqu'à Cluster details (Détails du cluster). En bas de cette section, recherchez le nom du groupe de paramètres sous Cluster parameter group (Groupe de paramètres de cluster).

   Using the AWS CLI

   Le AWS CLI code suivant détermine le paramètre qui régit votre cluster. Assurez-vous de remplacer `sample-cluster` par le nom de votre cluster.

   ```
   aws docdb describe-db-clusters \
       --db-cluster-identifier sample-cluster \
       --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
   ```

   Le résultat de cette opération ressemble à ce qui suit :

   ```
   [
          [
              "sample-cluster",
   ```

```
            "sample-parameter-group"
        ]
]
```

2. Déterminez la valeur du paramètre **`tls`** dans le groupe de paramètres de votre cluster.

   Using the AWS Management Console

   1. Dans le panneau de navigation, choisissez Groupes de paramètres.
   2. Dans la fenêtre Cluster parameter groups (Groupes de paramètres de cluster), sélectionnez votre groupe de paramètres de cluster.
   3. La page résultante affiche les paramètres de votre groupe de paramètres de cluster. Ici, vous pouvez voir la valeur du paramètre `tls`. Pour plus d'informations sur la modification de ce paramètre, veuillez consulter Modification des groupes de paramètres du cluster Amazon DocumentDB.

   Using the AWS CLI

   Vous pouvez utiliser la `describe-db-cluster-parameters` AWS CLI commande pour afficher les détails des paramètres de votre groupe de paramètres de cluster.

   - **`--describe-db-cluster-parameters`**— Pour répertorier tous les paramètres d'un groupe de paramètres et leurs valeurs.
     - **`--db-cluster-parameter-group name`** — Obligatoire. Nom de votre groupe de paramètres de cluster.

   ```
   aws docdb describe-db-cluster-parameters \
       --db-cluster-parameter-group-name sample-parameter-group
   ```

   Le résultat de cette opération ressemble à ce qui suit :

   ```
   {
           "Parameters": [
               {
                   "ParameterName": "profiler_threshold_ms",
                   "ParameterValue": "100",
                   "Description": "Operations longer than profiler_threshold_ms
    will be logged",
   ```

```
                "Source": "system",
                "ApplyType": "dynamic",
                "DataType": "integer",
                "AllowedValues": "50-2147483646",
                "IsModifiable": true,
                "ApplyMethod": "pending-reboot"
            },
            {
                "ParameterName": "tls",
                "ParameterValue": "disabled",
                "Description": "Config to enable/disable TLS",
                "Source": "user",
                "ApplyType": "static",
                "DataType": "string",
                "AllowedValues": "disabled,enabled,fips-140-3",
                "IsModifiable": true,
                "ApplyMethod": "pending-reboot"
            }
        ]
}
```

> **Note**
>
> Amazon DocumentDB prend en charge les points de terminaison FIPS 140-3 à partir des clusters Amazon DocumentDB 5.0 (moteur version 3.0.3727) dans les régions suivantes : ca-central-1, us-west-2, us-east-1, us-east-2, -1, -1. us-gov-east us-gov-west

Après avoir déterminé la valeur de votre paramètre `tls`, poursuivez la connexion de votre cluster en utilisant l'un des exemples de code des sections suivantes.

- Connexion avec TLS activé
- Connexion avec TLS désactivé

Connexion avec TLS activé

Pour afficher un exemple de code permettant de se connecter par programmation à un cluster Amazon DocumentDB compatible TLS, choisissez l'onglet correspondant à la langue que vous souhaitez utiliser.

Pour chiffrer les données en transit, téléchargez la clé publique pour Amazon DocumentDB `global-bundle.pem` nommée à l'aide de l'opération suivante.

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

Si votre application est sur Microsoft Windows et nécessite un fichier PKCS7, vous pouvez télécharger l'ensemble de certificats PKCS7. Cette solution groupée contient à la fois les certificats intermédiaires et racine à l'adresse https://truststore.pki.rds.amazonaws.com/global/global-bundle.p7b.

Python

Le code suivant montre comment se connecter à Amazon DocumentDB à l'aide de Python lorsque le protocole TLS est activé.

```
import pymongo
import sys

##Create a MongoDB client, open a connection to Amazon DocumentDB as a replica set
 and specify the read preference as secondary preferred
client = pymongo.MongoClient('mongodb://<sample-user>:<password>@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/?tls=true&tlsCAFile=global-
bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false')

##Specify the database to be used
db = client.sample_database

##Specify the collection to be used
col = db.sample_collection

##Insert a single document
col.insert_one({'hello':'Amazon DocumentDB'})

##Find the document that was previously written
x = col.find_one({'hello':'Amazon DocumentDB'})

##Print the result to the screen
print(x)

##Close the connection
client.close()
```

Node.js

Le code suivant montre comment se connecter à Amazon DocumentDB à l'aide de Node.js lorsque le protocole TLS est activé.

```
var MongoClient = require('mongodb').MongoClient

//Create a MongoDB client, open a connection to DocDB; as a replica set,
//  and specify the read preference as secondary preferred

var client = MongoClient.connect(
'mongodb://<sample-user>:<password>@sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017/sample-database?
tls=true&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false',
{
  tlsCAFile: `global-bundle.pem` //Specify the DocDB; cert
},
function(err, client) {
    if(err)
        throw err;

    //Specify the database to be used
    db = client.db('sample-database');

    //Specify the collection to be used
    col = db.collection('sample-collection');

    //Insert a single document
    col.insertOne({'hello':'Amazon DocumentDB'}, function(err, result){
      //Find the document that was previously written
      col.findOne({'hello':'DocDB;'}, function(err, result){
        //Print the result to the screen
        console.log(result);

        //Close the connection
        client.close()
      });
   });
});
```

PHP

Le code suivant montre comment se connecter à Amazon DocumentDB à l'aide de PHP lorsque le protocole TLS est activé.

```
<?php
//Include Composer's autoloader
require 'vendor/autoload.php';

$TLS_DIR = "/home/ubuntu/global-bundle.pem";

//Create a MongoDB client and open connection to Amazon DocumentDB
$client = new MongoDB\Client("mongodb://<sample-user>:<password>@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/?retryWrites=false", ["tls" =>
 "true", "tlsCAFile" => $TLS_DIR ]);

//Specify the database and collection to be used
$col = $client->sampledatabase->samplecollection;

//Insert a single document
$result = $col->insertOne( [ 'hello' => 'Amazon DocumentDB'] );

//Find the document that was previously written
$result = $col->findOne(array('hello' => 'Amazon DocumentDB'));

//Print the result to the screen
print_r($result);
?>
```

Go

Le code suivant montre comment se connecter à Amazon DocumentDB à l'aide de Go lorsque le protocole TLS est activé.

> **Note**
>
> À partir de la version 1.2.1, le pilote MongoDB Go n'utilisera que le premier certificat de serveur CA trouvé dans `sslcertificateauthorityfile`. L'exemple de code ci-dessous corrige cette limitation en ajoutant manuellement tous les certificats de serveur présents dans `sslcertificateauthorityfile` à une configuration TLS personnalisée utilisée lors de la création du client.

```
package main

import (
 "context"
 "fmt"
 "log"
 "time"

 "go.mongodb.org/mongo-driver/bson"
 "go.mongodb.org/mongo-driver/mongo"
 "go.mongodb.org/mongo-driver/mongo/options"

 "io/ioutil"
 "crypto/tls"
 "crypto/x509"
 "errors"
)

const (
 // Path to the AWS CA file
 caFilePath = "global-bundle.pem"

 // Timeout operations after N seconds
 connectTimeout  = 5
 queryTimeout    = 30
 username        = "<sample-user>"
 password        = "<password>"
 clusterEndpoint = "sample-cluster.node.us-east-1.docdb.amazonaws.com:27017"

 // Which instances to read from
 readPreference = "secondaryPreferred"

 connectionStringTemplate = "mongodb://%s:%s@%s/sample-database?
tls=true&replicaSet=rs0&readpreference=%s"
)

func main() {

 connectionURI := fmt.Sprintf(connectionStringTemplate, username, password,
 clusterEndpoint, readPreference)

 tlsConfig, err := getCustomTLSConfig(caFilePath)
 if err != nil {
```

```
 log.Fatalf("Failed getting TLS configuration: %v", err)
}

client, err :=
mongo.NewClient(options.Client().ApplyURI(connectionURI).SetTLSConfig(tlsConfig))
if err != nil {
 log.Fatalf("Failed to create client: %v", err)
}

ctx, cancel := context.WithTimeout(context.Background(),
connectTimeout*time.Second)
defer cancel()

err = client.Connect(ctx)
if err != nil {
 log.Fatalf("Failed to connect to cluster: %v", err)
}

// Force a connection to verify our connection string
err = client.Ping(ctx, nil)
if err != nil {
 log.Fatalf("Failed to ping cluster: %v", err)
}

fmt.Println("Connected to DocumentDB!")

collection := client.Database("sample-database").Collection("sample-collection")

ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
defer cancel()

res, err := collection.InsertOne(ctx, bson.M{"name": "pi", "value": 3.14159})
if err != nil {
 log.Fatalf("Failed to insert document: %v", err)
}

id := res.InsertedID
log.Printf("Inserted document ID: %s", id)

ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
defer cancel()

cur, err := collection.Find(ctx, bson.D{})
```

```
 if err != nil {
  log.Fatalf("Failed to run find query: %v", err)
 }
 defer cur.Close(ctx)

 for cur.Next(ctx) {
  var result bson.M
  err := cur.Decode(&result)
  log.Printf("Returned: %v", result)

  if err != nil {
   log.Fatal(err)
  }
 }

 if err := cur.Err(); err != nil {
  log.Fatal(err)
 }

}

func getCustomTLSConfig(caFile string) (*tls.Config, error) {
 tlsConfig := new(tls.Config)
 certs, err := ioutil.ReadFile(caFile)

 if err != nil {
  return tlsConfig, err
 }

 tlsConfig.RootCAs = x509.NewCertPool()
 ok := tlsConfig.RootCAs.AppendCertsFromPEM(certs)

 if !ok {
  return tlsConfig, errors.New("Failed parsing pem file")
 }

 return tlsConfig, nil
```

Java

Lorsque vous vous connectez à un cluster Amazon DocumentDB compatible TLS depuis une application Java, votre programme doit utiliser AWS le fichier d'autorité de certification (CA) fourni pour valider la connexion. Pour utiliser le certificat Amazon RDS CA, procédez comme suit :

1. Téléchargez le fichier Amazon RDS CA depuis https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem.
2. Créez un référentiel d'approbations avec le certificat d'autorité de certification (CA) contenu dans le fichier en exécutant les commandes suivantes. Assurez-vous de remplacer *<truststorePassword>* par une autre valeur. Si vous accédez à un référentiel d'approbations qui contient à la fois l'ancien certificat d'autorité de certification (`rds-ca-2015-root.pem`) et le nouveau certificat d'autorité de certification (`rds-ca-2019-root.pem`), vous pouvez importer le bundle de certificats dans le référentiel d'approbations.

   Voici un exemple de scripting shell qui importe le lot de certificats vers un magasin d'approbations sur un système d'exploitation Linux. Dans les exemples suivants, remplacez chaque *espace réservé pour l'entrée utilisateur* par vos propres informations. Plus particulièrement, partout où le répertoire d'exemple « *mydir* » se trouve dans le script, remplacez-le par un répertoire que vous avez créé pour cette tâche.

```
mydir=/tmp/certs
truststore=${mydir}/rds-truststore.jks
storepassword=<truststorePassword>

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem" >
 ${mydir}/global-bundle.pem
awk 'split_after == 1 {n++;split_after=0} /-----END CERTIFICATE-----/
 {split_after=1}{print > "rds-ca-" n ".pem"}' < ${mydir}/global-bundle.pem

for CERT in rds-ca-*; do
  alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /
Subject:/; s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -
keystore ${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep
 Alias | cut -d " " -f3- | while read alias
do
```

```
   expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword}
 -alias "${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print
 "$1\n"; }'`
   echo " Certificate ${alias} expires in '$expiry'"
done
```

Voici un exemple de scripting shell qui importe le lot de certificats vers un magasin d'approbations sur macOS.

```
mydir=/tmp/certs
truststore=${mydir}/rds-truststore.jks
storepassword=<truststorePassword>

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem" >
 ${mydir}/global-bundle.pem
split -p "-----BEGIN CERTIFICATE-----" ${mydir}/global-bundle.pem rds-ca-

for CERT in rds-ca-*; do
  alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /
Subject:/; s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -
keystore ${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep
 Alias | cut -d " " -f3- | while read alias
do
   expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword}
 -alias "${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print
 "$1\n"; }'`
   echo " Certificate ${alias} expires in '$expiry'"
done
```

3. Utilisez-le `keystore` dans votre programme en définissant les propriétés système suivantes dans votre application avant d'établir une connexion au cluster Amazon DocumentDB.

```
javax.net.ssl.trustStore: <truststore>
javax.net.ssl.trustStorePassword: <truststorePassword>
```

4. Le code suivant montre comment se connecter à Amazon DocumentDB à l'aide de Java lorsque le protocole TLS est activé.

```
package com.example.documentdb;

import com.mongodb.client.*;
import org.bson.Document;

public final class Test {
    private Test() {
    }
    public static void main(String[] args) {

        String template = "mongodb://%s:%s@%s/sample-database?
ssl=true&replicaSet=rs0&readpreference=%s";
        String username = "<sample-user>";
        String password = "<password>";
        String clusterEndpoint = "sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
        String readPreference = "secondaryPreferred";
        String connectionString = String.format(template, username, password,
 clusterEndpoint, readPreference);

        String truststore = "<truststore>";
        String truststorePassword = "<truststorePassword>";

        System.setProperty("javax.net.ssl.trustStore", truststore);
        System.setProperty("javax.net.ssl.trustStorePassword",
 truststorePassword);

        MongoClient mongoClient = MongoClients.create(connectionString);

        MongoDatabase testDB = mongoClient.getDatabase("sample-database");
        MongoCollection<Document> numbersCollection =
 testDB.getCollection("sample-collection");

        Document doc = new Document("name", "pi").append("value", 3.14159);
        numbersCollection.insertOne(doc);
```

```
        MongoCursor<Document> cursor = numbersCollection.find().iterator();
        try {
            while (cursor.hasNext()) {
                System.out.println(cursor.next().toJson());
            }
        } finally {
            cursor.close();
        }

    }
}
```

C# / .NET

Le code suivant montre comment se connecter à Amazon DocumentDB à l'aide de C#/.NET lorsque le protocole TLS est activé.

```
using System;
using System.Text;
using System.Linq;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;
using System.Net.Security;
using MongoDB.Driver;
using MongoDB.Bson;

namespace DocDB
{
    class Program
    {
        static void Main(string[] args)
        {
            string template = "mongodb://{0}:{1}@{2}/sampledatabase?
tls=true&replicaSet=rs0&readpreference={3}";
            string username = "<sample-user>";
            string password = "<password>";
            string readPreference = "secondaryPreferred";
            string clusterEndpoint="sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
            string connectionString = String.Format(template, username, password,
 clusterEndpoint, readPreference);
```

```
            string pathToCAFile = "<PATH/global-bundle.p7b_file>";

            // ADD CA certificate to local trust store
            // DO this once - Maybe when your service starts
            X509Store localTrustStore = new X509Store(StoreName.Root);
            X509Certificate2Collection certificateCollection = new
 X509Certificate2Collection();
            certificateCollection.Import(pathToCAFile);
            try
            {
                localTrustStore.Open(OpenFlags.ReadWrite);
                localTrustStore.AddRange(certificateCollection);
            }
            catch (Exception ex)
            {
                Console.WriteLine("Root certificate import failed: " + ex.Message);
                throw;
            }
            finally
            {
                localTrustStore.Close();
            }

            var settings = MongoClientSettings.FromUrl(new
 MongoUrl(connectionString));
            var client = new MongoClient(settings);

            var database = client.GetDatabase("sampledatabase");
            var collection =
 database.GetCollection<BsonDocument>("samplecollection");
            var docToInsert = new BsonDocument { { "pi", 3.14159 } };
            collection.InsertOne(docToInsert);
        }
    }
}
```

mongo shell

Le code suivant montre comment se connecter à Amazon DocumentDB et l'interroger à l'aide du shell mongo lorsque le protocole TLS est activé.

1. Connectez-vous à Amazon DocumentDB avec le shell mongo. Si vous utilisez une version de shell mongo antérieure à 4.2, utilisez le code suivant pour vous connecter.

   ```
   mongo --ssl --host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 --
   sslCAFile global-bundle.pem --username <sample-user> --password <password>
   ```

   Si vous utilisez une version égale ou supérieure à 4.2, utilisez le code suivant pour vous connecter. Les écritures réessayables ne sont pas prises en charge dans AWS DocumentDB. Exception : si vous utilisez le shell mongo, n'incluez la `retryWrites=false` commande dans aucune chaîne de code. Par défaut, les écritures réessayables sont désactivées. L'inclusion `retryWrites=false` peut entraîner un échec dans les commandes de lecture normales.

   ```
   mongo --tls --host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 --
   tlsCAFile global-bundle.pem --username <sample-user> --password <password>
   ```

2. Insérez un document unique.

   ```
   db.myTestCollection.insertOne({'hello':'Amazon DocumentDB'})
   ```

3. Recherchez le document qui a été inséré précédemment.

   ```
   db.myTestCollection.find({'hello':'Amazon DocumentDB'})
   ```

R

Le code suivant montre comment se connecter à Amazon DocumentDB avec R à l'aide de mongolite (https://jeroen.github.io/mongolite/) lorsque le protocole TLS est activé.

```
#Include the mongolite library.
library(mongolite)

mongourl <- paste("mongodb://<sample-user>:<password>@sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017/test2?ssl=true&",
          "readPreference=secondaryPreferred&replicaSet=rs0", sep="")

#Create a MongoDB client, open a connection to Amazon DocumentDB as a replica
#   set and specify the read preference as secondary preferred
client <-  mongo(url = mongourl, options = ssl_options(weak_cert_validation = F, ca
 ="<PATH/global-bundle.pem>"))
```

```
#Insert a single document
str <- c('{"hello" : "Amazon DocumentDB"}')
client$insert(str)

#Find the document that was previously written
client$find()
```

Ruby

Le code suivant montre comment se connecter à Amazon DocumentDB avec Ruby lorsque le protocole TLS est activé.

```
require 'mongo'
require 'neatjson'
require 'json'
client_host = 'mongodb://sample-cluster.node.us-east-1.docdb.amazonaws.com:27017'
client_options = {
   database: 'test',
   replica_set: 'rs0',
   read: {:secondary_preferred => 1},
   user: '<sample-user>',
   password: '<password>',
   ssl: true,
   ssl_verify: true,
   ssl_ca_cert: <'PATH/global-bundle.pem'>,
   retry_writes: false
}


begin
   ##Create a MongoDB client, open a connection to Amazon DocumentDB as a
   ##   replica set and specify the read preference as secondary preferred
   client = Mongo::Client.new(client_host, client_options)

   ##Insert a single document
   x = client[:test].insert_one({"hello":"Amazon DocumentDB"})

   ##Find the document that was previously written
   result = client[:test].find()

   #Print the document
   result.each do |document|
```

```
      puts JSON.neat_generate(document)
   end
end

#Close the connection
client.close
```

Connexion avec TLS désactivé

Pour afficher un exemple de code permettant de se connecter par programmation à un cluster Amazon DocumentDB désactivé par TLS, choisissez l'onglet correspondant à la langue que vous souhaitez utiliser.

Python

Le code suivant montre comment se connecter à Amazon DocumentDB à l'aide de Python lorsque le protocole TLS est désactivé.

```
## Create a MongoDB client, open a connection to Amazon DocumentDB as a replica set
 and specify the read preference as secondary preferred

import pymongo
import sys

client = pymongo.MongoClient('mongodb://<sample-user>:<password>@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false')

##Specify the database to be used
db = client.sample_database

##Specify the collection to be used
col = db.sample_collection

##Insert a single document
col.insert_one({'hello':'Amazon DocumentDB'})

##Find the document that was previously written
x = col.find_one({'hello':'Amazon DocumentDB'})

##Print the result to the screen
print(x)
```

```
##Close the connection
client.close()
```

Node.js

Le code suivant montre comment se connecter à Amazon DocumentDB à l'aide de Node.js lorsque le protocole TLS est désactivé.

```
var MongoClient = require('mongodb').MongoClient;

//Create a MongoDB client, open a connection to Amazon DocumentDB as a replica set,
//  and specify the read preference as secondary preferred
var client = MongoClient.connect(
'mongodb://<sample-user>:<password>@sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017/sample-database?
replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false',
{
  useNewUrlParser: true
},

function(err, client) {
    if(err)
        throw err;
    //Specify the database to be used
    db = client.db('sample-database');

    //Specify the collection to be used
    col = db.collection('sample-collection');

    //Insert a single document
    col.insertOne({'hello':'Amazon DocumentDB'}, function(err, result){
      //Find the document that was previously written
      col.findOne({'hello':'Amazon DocumentDB'}, function(err, result){
        //Print the result to the screen
        console.log(result);

        //Close the connection
        client.close()
      });
   });
});
```

PHP

Le code suivant montre comment se connecter à Amazon DocumentDB à l'aide de PHP lorsque le protocole TLS est désactivé.

```
<?php
//Include Composer's autoloader
require 'vendor/autoload.php';

//Create a MongoDB client and open connection to Amazon DocumentDB
$client = new MongoDB\Client("mongodb://<sample-user>:<password>@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/?retryWrites=false");

//Specify the database and collection to be used
$col = $client->sampledatabase->samplecollection;

//Insert a single document
$result = $col->insertOne( [ 'hello' => 'Amazon DocumentDB'] );

//Find the document that was previously written
$result = $col->findOne(array('hello' => 'Amazon DocumentDB'));

//Print the result to the screen
print_r($result);
?>
```

Go

Le code suivant montre comment se connecter à Amazon DocumentDB à l'aide de Go lorsque le protocole TLS est désactivé.

```
package main

import (
 "context"
 "fmt"
 "log"
 "time"

 "go.mongodb.org/mongo-driver/bson"
 "go.mongodb.org/mongo-driver/mongo"
 "go.mongodb.org/mongo-driver/mongo/options"
)
```

```
const (
 // Timeout operations after N seconds
 connectTimeout  = 5
 queryTimeout    = 30
 username        = "<sample-user>"
 password        = "<password>"
 clusterEndpoint = "sample-cluster.node.us-east-1.docdb.amazonaws.com:27017"

 // Which instances to read from
 readPreference           = "secondaryPreferred"
 connectionStringTemplate = "mongodb://%s:%s@%s/sample-database?
replicaSet=rs0&readpreference=%s"
)

func main() {

 connectionURI := fmt.Sprintf(connectionStringTemplate, username, password,
 clusterEndpoint, readPreference)

 client, err := mongo.NewClient(options.Client().ApplyURI(connectionURI))
 if err != nil {
  log.Fatalf("Failed to create client: %v", err)
 }

 ctx, cancel := context.WithTimeout(context.Background(),
 connectTimeout*time.Second)
 defer cancel()

 err = client.Connect(ctx)
 if err != nil {
  log.Fatalf("Failed to connect to cluster: %v", err)
 }

 // Force a connection to verify our connection string
 err = client.Ping(ctx, nil)
 if err != nil {
  log.Fatalf("Failed to ping cluster: %v", err)
 }

 fmt.Println("Connected to DocumentDB!")

 collection := client.Database("sample-database").Collection("sample-collection")
```

```
 ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
 defer cancel()

 res, err := collection.InsertOne(ctx, bson.M{"name": "pi", "value": 3.14159})
 if err != nil {
  log.Fatalf("Failed to insert document: %v", err)
 }

 id := res.InsertedID
 log.Printf("Inserted document ID: %s", id)

 ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
 defer cancel()

 cur, err := collection.Find(ctx, bson.D{})

 if err != nil {
  log.Fatalf("Failed to run find query: %v", err)
 }
 defer cur.Close(ctx)

 for cur.Next(ctx) {
  var result bson.M
  err := cur.Decode(&result)
  log.Printf("Returned: %v", result)

  if err != nil {
   log.Fatal(err)
  }
 }

 if err := cur.Err(); err != nil {
  log.Fatal(err)
 }

}
```

Java

Le code suivant montre comment se connecter à Amazon DocumentDB à l'aide de Java lorsque le protocole TLS est désactivé.

```
package com.example.documentdb;
```

```
import com.mongodb.MongoClient;
import com.mongodb.MongoClientURI;
import com.mongodb.ServerAddress;
import com.mongodb.MongoException;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.MongoCollection;
import org.bson.Document;


public final class Main {
    private Main() {
    }
    public static void main(String[] args) {

        String template = "mongodb://%s:%s@%s/sample-database?
replicaSet=rs0&readpreference=%s";
        String username = "<sample-user>";
        String password = "<password>";
        String clusterEndpoint = "sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
        String readPreference = "secondaryPreferred";
        String connectionString = String.format(template, username, password,
 clusterEndpoint, readPreference);

        MongoClientURI clientURI = new MongoClientURI(connectionString);
        MongoClient mongoClient = new MongoClient(clientURI);

        MongoDatabase testDB = mongoClient.getDatabase("sample-database");
        MongoCollection<Document> numbersCollection = testDB.getCollection("sample-
collection");

        Document doc = new Document("name", "pi").append("value", 3.14159);
        numbersCollection.insertOne(doc);

        MongoCursor<Document> cursor = numbersCollection.find().iterator();
        try {
            while (cursor.hasNext()) {
                System.out.println(cursor.next().toJson());
            }
        } finally {
            cursor.close();
        }
```

```
    }
}
```

C# / .NET

Le code suivant montre comment se connecter à Amazon DocumentDB à l'aide de C#/.NET lorsque le protocole TLS est désactivé.

```
using System;
using System.Text;
using System.Linq;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;
using System.Net.Security;
using MongoDB.Driver;
using MongoDB.Bson;

namespace CSharpSample
{
   class Program
    {
       static void Main(string[] args)
        {
           string template = "mongodb://{0}:{1}@{2}/sampledatabase?
replicaSet=rs0&readpreference={3}";
           string username = "<sample-user>";
           string password = "<password>";
           string clusterEndpoint = "sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
           string readPreference = "secondaryPreferred";
           string connectionString = String.Format(template, username, password,
 clusterEndpoint, readPreference);

           var settings = MongoClientSettings.FromUrl(new
 MongoUrl(connectionString));
           var client = new MongoClient(settings);

           var database = client.GetDatabase("sampledatabase");
           var collection =
 database.GetCollection<BsonDocument>("samplecollection");
           var docToInsert = new BsonDocument { { "pi", 3.14159 } };
```

```
            collection.InsertOne(docToInsert);
        }
    }
}
```

mongo shell

Le code suivant montre comment se connecter à Amazon DocumentDB et l'interroger à l'aide du shell mongo lorsque le protocole TLS est désactivé.

1. Connectez-vous à Amazon DocumentDB avec le shell mongo.

```
mongo --host mycluster.node.us-east-1.docdb.amazonaws.com:27017 --
username <sample-user> --password <password>
```

2. Insérez un document unique.

```
db.myTestCollection.insertOne({'hello':'Amazon DocumentDB'})
```

3. Recherchez le document qui a été inséré précédemment.

```
db.myTestCollection.find({'hello':'Amazon DocumentDB'})
```

R

Le code suivant montre comment se connecter à Amazon DocumentDB avec R à l'aide de mongolite (https://jeroen.github.io/mongolite/) lorsque le protocole TLS est désactivé.

```
#Include the mongolite library.
library(mongolite)

#Create a MongoDB client, open a connection to Amazon DocumentDB as a replica
#   set and specify the read preference as secondary preferred
client <- mongo(url = "mongodb://<sample-user>:<password>@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/sample-database?
readPreference=secondaryPreferred&replicaSet=rs0")

##Insert a single document
str <- c('{"hello" : "Amazon DocumentDB"}')
client$insert(str)
```

```
##Find the document that was previously written
client$find()
```

Ruby

Le code suivant montre comment se connecter à Amazon DocumentDB avec Ruby lorsque le protocole TLS est désactivé.

```
require 'mongo'
require 'neatjson'
require 'json'
client_host = 'mongodb://sample-cluster.node.us-east-1.docdb.amazonaws.com:27017'
client_options = {
   database: 'test',
   replica_set: 'rs0',
   read: {:secondary_preferred => 1},
   user: '<sample-user>',
   password: '<password>',
   retry_writes: false
}

begin
   ##Create a MongoDB client, open a connection to Amazon DocumentDB as a
   ##   replica set and specify the read preference as secondary preferred
   client = Mongo::Client.new(client_host, client_options)

   ##Insert a single document
   x = client[:test].insert_one({"hello":"Amazon DocumentDB"})

   ##Find the document that was previously written
   result = client[:test].find()

   #Print the document
   result.each do |document|
      puts JSON.neat_generate(document)
   end
end

#Close the connection
client.close
```

Utilisation de Change Streams avec Amazon DocumentDB

La fonctionnalité de flux de modifications d'Amazon DocumentDB (compatible avec MongoDB) fournit une séquence chronologique des événements de modification qui se produisent dans les collections de votre cluster. Vous pouvez lire des événements à partir d'un flux de modifications afin d'implémenter de nombreux cas d'utilisation différents, notamment les suivants :

- Notification de modification
- Recherche en texte intégral avec Amazon OpenSearch Service (OpenSearch Service)
- Analyses avec Amazon Redshift

Les applications peuvent utiliser les flux de modifications pour souscrire à tous les changements de données dans des collections individuelles. Les événements de flux de modifications sont ordonnés au fur et à mesure qu'ils se produisent sur le cluster et sont stockés pendant trois heures (par défaut) après l'enregistrement de l'événement. La période de conservation peut être prolongée jusqu'à 7 jours en utilisant le`change_stream_log_retention_duration`paramètre. Pour modifier la période de conservation du flux de modifications, consultezModification de la durée de conservation du journal du flux de modifications.

Rubriques

Opérations prises en charge

Amazon DocumentDB prend en charge les opérations suivantes pour les flux de modifications :

- Tous les événements de changement pris en charge dans MongoDB`db.collection.watch()`,`db.watch()`et`client.watch()`API.
- Recherche complète de documents pour les mises à jour.
- Étapes d'agrégation :`$match`,`$project`,`$redact`, et`$addFields`et`$replaceRoot`.
- Reprise d'un flux de modifications à partir d'un jeton de CV
- Reprise d'un flux de modifications à partir d'un horodatage en utilisant`startAtOperation`(applicable à Amazon DocumentDB v4.0+)

Facturation

La fonctionnalité de flux de modifications d'Amazon DocumentDB est désactivée par défaut et n'entraîne aucun frais supplémentaire tant qu'elle n'est pas activée. L'utilisation de flux de modifications dans un cluster entraîne des coûts supplémentaires en lecture et en écriture, ainsi que des coûts de stockage. Vous pouvez utiliser le`modifyChangeStreams`Opération d'API permettant d'activer cette fonctionnalité pour votre cluster. Pour plus d'informations sur les tarifs, voirTarification d'Amazon DocumentDB.

Limites

Les flux de modifications présentent les limites suivantes dans Amazon DocumentDB :

- Les flux de modifications ne peuvent être ouverts qu'à partir d'une connexion à l'instance principale d'un cluster Amazon DocumentDB. La lecture à partir de flux de modifications sur une instance de réplica n'est actuellement pas prise en charge. Lorsque vous appelez l'opération d'API `watch()`, vous devez spécifier une préférence de lecture **`primary`** pour vous assurer que toutes les lectures sont dirigées vers l'instance principale (veuillez consulter la section Exemple).
- Les événements écrits dans un flux de modifications pour une collection sont disponibles pendant 7 jours maximum (la valeur par défaut est de 3 heures). Les données du flux de modifications sont supprimées après la fenêtre de durée de conservation du journal, même si aucune nouvelle modification n'est survenue.
- Une opération d'écriture longue en cours d'exécution sur une collection comme `updateMany` ou `deleteMany` peut bloquer temporairement l'écriture des événements de flux de modifications jusqu'à ce que l'opération d'écriture longue en cours d'exécution soit terminée.
- Amazon DocumentDB ne prend pas en charge le journal des opérations MongoDB (`oplog`).
- Avec Amazon DocumentDB, vous devez activer explicitement les flux de modifications sur une collection donnée.

- Si la taille totale d'un événement de flux de modifications (y compris les données de modification et le document complet, le cas échéant) est supérieure à `16 MB`, le client rencontre un échec de lecture sur les flux de modification.
- Le pilote Ruby n'est actuellement pas pris en charge lors de l'utilisation`db.watch()`et`client.watch()`avec Amazon DocumentDB v3.6.

Activation des flux de modifications

Vous pouvez activer les flux de modifications Amazon DocumentDB pour toutes les collections d'une base de données donnée, ou uniquement pour certaines collections. Voici des exemples de la façon d'activer les flux de changement pour différents cas d'utilisation en utilisant le shell mongo. Les chaînes vides sont traitées comme des caractères génériques lors de la spécification des noms de base de données et de collections.

```
//Enable change streams for the collection "foo" in database "bar"
db.adminCommand({modifyChangeStreams: 1,
    database: "bar",
    collection: "foo",
    enable: true});
```

```
//Disable change streams on collection "foo" in database "bar"
db.adminCommand({modifyChangeStreams: 1,
    database: "bar",
    collection: "foo",
    enable: false});
```

```
//Enable change streams for all collections in database "bar"
db.adminCommand({modifyChangeStreams: 1,
    database: "bar",
    collection: "",
    enable: true});
```

```
//Enable change streams for all collections in all databases in a cluster
db.adminCommand({modifyChangeStreams: 1,
    database: "",
    collection: "",
    enable: true});
```

Les flux de modifications seront activés pour une collection si l'un des éléments suivants est vrai :

- La base de données et la collection sont explicitement activées.
- La base de données contenant la collection est activée.
- Toutes les bases de données sont activées.

Le fait de supprimer une collection d'une base de données ne désactive pas les flux de modification pour cette collection si la base de données parent a également activé les flux de modification ou si toutes les bases de données du cluster sont activées. Si une nouvelle collection est créée avec le même nom que la collection supprimée, les flux de modifications seront activés pour cette collection.

Vous pouvez répertorier tous les flux de modifications activés de votre cluster à l'aide de l'étape du pipeline d'agrégation `$listChangeStreams`. Toutes les étapes d'agrégation prises en charge par Amazon DocumentDB peuvent être utilisées dans le pipeline pour un traitement supplémentaire. Si une collection précédemment activée a été désactivée, elle n'apparaît pas dans la sortie `$listChangeStreams`.

```
//List all databases and collections with change streams enabled
cursor = new DBCommandCursor(db,
    db.runCommand(
        {aggregate: 1,
        pipeline: [{$listChangeStreams: 1}],
        cursor:{}}));
```

```
//List of all databases and collections with change streams enabled
{ "database" : "test", "collection" : "foo" }
{ "database" : "bar", "collection" : "" }
{ "database" : "", "collection" : "" }
```

```
//Determine if the database “bar” or collection “bar.foo” have change streams enabled
cursor = new DBCommandCursor(db,
  db.runCommand(
      {aggregate: 1,
       pipeline: [{$listChangeStreams: 1},
                  {$match: {$or: [{database: "bar", collection: "foo"},
                                  {database: "bar", collection: ""},
                                  {database: "", collection: ""}]}}
                 ],
      cursor:{}}));
```

Exemple : Utilisation de flux de modifications avec Python

Voici un exemple d'utilisation d'un flux de modifications Amazon DocumentDB avec Python au niveau de la collection.

```
import os
import sys
from pymongo import MongoClient, ReadPreference

username = "DocumentDBusername"
password = <Insert your password>

clusterendpoint = "DocumentDBClusterEndpoint”
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
 tlsCAFile='global-bundle.pem')

db = client['bar']

#While ‘Primary’ is the default read preference, here we give an example of
#how to specify the required read preference when reading the change streams
coll = db.get_collection('foo', read_preference=ReadPreference.PRIMARY)
#Create a stream object
stream = coll.watch()
#Write a new document to the collection to generate a change event
coll.insert_one({'x': 1})
#Read the next change event from the stream (if any)
print(stream.try_next())

"""
Expected Output:
{'_id': {'_data': '015daf94f600000002010000000200009025'},
'clusterTime': Timestamp(1571788022, 2),
'documentKey': {'_id': ObjectId('5daf94f6ea258751778163d6')},
'fullDocument': {'_id': ObjectId('5daf94f6ea258751778163d6'), 'x': 1},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'insert'}
"""

#A subsequent attempt to read the next change event returns nothing, as there are no
 new changes
print(stream.try_next())

"""
```

```
Expected Output:
None
"""

#Generate a new change event by updating a document
result = coll.update_one({'x': 1}, {'$set': {'x': 2}})
print(stream.try_next())

"""
Expected Output:
{'_id': {'_data': '015daf99d40000000101000000100009025'},
'clusterTime': Timestamp(1571789268, 1),
'documentKey': {'_id': ObjectId('5daf9502ea258751778163d7')},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'update',
'updateDescription': {'removedFields': [], 'updatedFields': {'x': 2}}}
"""
```

Voici un exemple d'utilisation d'un flux de modifications Amazon DocumentDB avec Python au niveau de la base de données.

```
import os
import sys
from pymongo import MongoClient

username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint”
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
 tlsCAFile='global-bundle.pem')

db = client['bar']
#Create a stream object
stream = db.watch()
coll = db.get_collection('foo')
#Write a new document to the collection foo to generate a change event
coll.insert_one({'x': 1})

#Read the next change event from the stream (if any)
print(stream.try_next())

"""
Expected Output:
```

```
{'_id': {'_data': '015daf94f600000002010000000200009025'},
'clusterTime': Timestamp(1571788022, 2),
'documentKey': {'_id': ObjectId('5daf94f6ea258751778163d6')},
'fullDocument': {'_id': ObjectId('5daf94f6ea258751778163d6'), 'x': 1},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'insert'}
"""
#A subsequent attempt to read the next change event returns nothing, as there are no
 new changes
print(stream.try_next())

"""
Expected Output:
None
"""

coll = db.get_collection('foo1')

#Write a new document to another collection to generate a change event
coll.insert_one({'x': 1})
print(stream.try_next())

"""
Expected Output: Since the change stream cursor was the database level you can see
 change events from different collections in the same database
{'_id': {'_data': '015daf94f600000002010000000200009025'},
'clusterTime': Timestamp(1571788022, 2),
'documentKey': {'_id': ObjectId('5daf94f6ea258751778163d6')},
'fullDocument': {'_id': ObjectId('5daf94f6ea258751778163d6'), 'x': 1},
'ns': {'coll': 'foo1', 'db': 'bar'},
'operationType': 'insert'}
"""
```

Recherche complète de document

L'événement de modification de mise à jour n'inclut pas le document complet, mais uniquement la modification qui a été effectuée. Si votre cas d'utilisation nécessite le document complet affecté par une mise à jour, vous pouvez activer la recherche complète de documents lors de l'ouverture du flux.

Le document `fullDocument` d'un événement de flux de modification de mise à jour représente la version la plus récente du document mis à jour au moment de la recherche de document. Si des

modifications se sont produites entre l'opération de mise à jour et la recherche `fullDocument`, le document `fullDocument` peut ne pas représenter l'état du document au moment de la mise à jour.

```
#Create a stream object with update lookup enabled
stream = coll.watch(full_document='updateLookup')

#Generate a new change event by updating a document
result = coll.update_one({'x': 2}, {'$set': {'x': 3}})

stream.try_next()

#Output:
{'_id': {'_data': '015daf9b7c000000010100000100009025'},
'clusterTime': Timestamp(1571789692, 1),
'documentKey': {'_id': ObjectId('5daf9502ea258751778163d7')},
'fullDocument': {'_id': ObjectId('5daf9502ea258751778163d7'), 'x': 3},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'update',
'updateDescription': {'removedFields': [], 'updatedFields': {'x': 3}}}
```

Reprise d'un flux de modifications

Vous pouvez reprendre un flux de modifications ultérieurement à l'aide d'un jeton de reprise, qui est égal au champ `_id`du dernier document d'événement de modification récupéré.

```
import os
import sys
from pymongo import MongoClient

username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint”
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
 tlsCAFile='global-bundle.pem', retryWrites='false')

db = client['bar']
coll = db.get_collection('foo')
#Create a stream object
stream = db.watch()
coll.update_one({'x': 1}, {'$set': {'x': 4}})
event = stream.try_next()
token = event['_id']
```

```
print(token)

"""
Output: This is the resume token that we will later us to resume the change stream
{'_data': '015daf9c5b0000000101000000100009025'}
"""
#Python provides a nice shortcut for getting a stream's resume token
print(stream.resume_token)

"""
Output
{'_data': '015daf9c5b0000000101000000100009025'}
"""
#Generate a new change event by updating a document
result = coll.update_one({'x': 4}, {'$set': {'x': 5}})
#Generate another change event by inserting a document
result = coll.insert_one({'y': 5})
#Open a stream starting after the selected resume token
stream = db.watch(full_document='updateLookup', resume_after=token)
#Our first change event is the update with the specified _id
print(stream.try_next())

"""
#Output: Since we are resuming the change stream from the resume token, we will see all
 events after the first update operation. In our case, the change stream will resume
 from the update operation {x:5}

{'_id': {'_data': '015f7e8f0c0000000601000000006000fe038'},
'operationType': 'update',
'clusterTime': Timestamp(1602129676, 6),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e8f0ac423bafbfd9adba2')},
'fullDocument': {'_id': ObjectId('5f7e8f0ac423bafbfd9adba2'), 'x': 5},
'updateDescription': {'updatedFields': {'x': 5}, 'removedFields': []}}
"""
#Followed by the insert
print(stream.try_next())

"""
#Output:
{'_id': {'_data': '015f7e8f0c0000000701000000007000fe038'},
'operationType': 'insert',
'clusterTime': Timestamp(1602129676, 7),
'ns': {'db': 'bar', 'coll': 'foo'},
```

```
'documentKey': {'_id': ObjectId('5f7e8f0cbf8c233ed577eb94')},
'fullDocument': {'_id': ObjectId('5f7e8f0cbf8c233ed577eb94'), 'y': 5}}
"""
```

Reprise d'un flux de modifications avec`startAtOperationTime`

Vous pouvez reprendre un flux de modifications ultérieurement à partir d'un horodatage spécifique en utilisant`startAtOperationTime`.

> **Note**
>
> La capacité d'utiliser`startAtOperationTime`est disponible dans Amazon DocumentDB 4.0+. Lors de l'utilisation`startAtOperationTime`, le curseur du flux de modifications renverra uniquement les modifications survenues à ou après l'horodatage spécifié. Le`startAtOperationTime`et`resumeAfter`les commandes s'excluent mutuellement et ne peuvent donc pas être utilisées ensemble.

```
import os
import sys
from pymongo import MongoClient

username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint”
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
 tlsCAFile='rds-root-ca-2020.pem',retryWrites='false')
db = client['bar']
coll = db.get_collection('foo')
#Create a stream object
stream = db.watch()
coll.update_one({'x': 1}, {'$set': {'x': 4}})
event = stream.try_next()
timestamp = event['clusterTime']
print(timestamp)
"""
Output
Timestamp(1602129114, 4)
"""
#Generate a new change event by updating a document
result = coll.update_one({'x': 4}, {'$set': {'x': 5}})
```

```
result = coll.insert_one({'y': 5})
#Generate another change event by inserting a document
#Open a stream starting after specified time stamp

stream = db.watch(start_at_operation_time=timestamp)
print(stream.try_next())

"""
#Output: Since we are resuming the change stream at the time stamp of our first update
 operation (x:4), the change stream cursor will point to that event
{'_id': {'_data': '015f7e941a0000000301000000003000fe038'},
'operationType': 'update',
'clusterTime': Timestamp(1602130970, 3),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e9417c423bafbfd9adbb1')},
'updateDescription': {'updatedFields': {'x': 4}, 'removedFields': []}}
"""

print(stream.try_next())
"""
#Output: The second event will be the subsequent update operation (x:5)
{'_id': {'_data': '015f7e9502000000050100000005000fe038'},
'operationType': 'update',
'clusterTime': Timestamp(1602131202, 5),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e94ffc423bafbfd9adbb2')},
'updateDescription': {'updatedFields': {'x': 5}, 'removedFields': []}}
"""

print(stream.try_next())

"""
#Output: And finally the last event will be the insert operation (y:5)
{'_id': {'_data': '015f7e9502000000060100000006000fe038'},
'operationType': 'insert',
'clusterTime': Timestamp(1602131202, 6),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e95025c4a569e0f6dde92')},
'fullDocument': {'_id': ObjectId('5f7e95025c4a569e0f6dde92'), 'y': 5}}
"""
```

Transactions dans les flux de changement

Les événements du flux de modifications ne contiendront pas d'événements provenant de transactions non validées et/ou abandonnées. Par exemple, si vous commencez une transaction avec un`INSERT`opération et un`UPDATE`fonctionnement et. Si votre`INSERT`l'opération réussit, mais le`UPDATE`l'opération échoue, la transaction sera annulée. Cette transaction ayant été annulée, votre flux de modifications ne contiendra aucun événement lié à cette transaction.

Modification de la durée de conservation du journal du flux de modifications

Vous pouvez modifier la durée de conservation du journal du flux de modifications pour qu'elle soit comprise entre 1 heure et 7 jours à l'aide duAWS Management Consoleou leAWS CLI.

Using the AWS Management Console

Pour modifier la durée de conservation du journal du flux de modifications

1. Connectez-vous auAWS Management Console, et ouvrez la console Amazon DocumentDB à l'adressehttps://console.aws.amazon.com/docdb.
2. Dans le panneau de navigation, choisissez Groupes de paramètres.

   > **Tip**
   >
   > Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

3. Dans le volet Parameter groups (Groupes de paramètres), choisissez le groupe de paramètres de cluster associé à votre cluster. Pour identifier le groupe de paramètres de cluster associé à votre cluster, veuillez consulter Déterminer le groupe de paramètres d'un cluster Amazon DocumentDB.
4. La page qui s’affiche contient les paramètres et leurs détails correspondants pour votre groupe de paramètres de cluster. Sélectionnez le paramètre `change_stream_log_retention_duration`.
5. En haut à droite de la page, choisissez Edit (Modifier) pour modifier la valeur du paramètre. Le`change_stream_log_retention_duration`le paramètre peut être modifié pour être compris entre 1 heure et 7 jours.

6. Effectuez votre modification, puis choisissez Modify cluster parameter (Modifier le paramètre de cluster) pour enregistrer les modifications. Pour ignorer vos modifications, choisissez Annuler.

Using the AWS CLI

Pour modifier le paramètre `change_stream_log_retention_duration` de votre groupe de paramètres de cluster, utilisez l'opération `modify-db-cluster-parameter-group` avec les paramètres suivants :

- **`--db-cluster-parameter-group-name`** — Obligatoire. Nom du groupe de paramètres de cluster que vous voulez modifier. Pour identifier le groupe de paramètres de cluster associé à votre cluster, veuillez consulter Déterminer le groupe de paramètres d'un cluster Amazon DocumentDB.
- **`--parameters`** — Obligatoire. Paramètre que vous modifiez. Chaque saisie de paramètre doit inclure ce qui suit :
  - **`ParameterName`**— Le nom du paramètre que vous êtes en train de modifier. Dans ce cas, il s'agit de `change_stream_log_retention_duration`
  - **`ParameterValue`**— La nouvelle valeur de ce paramètre.
  - **`ApplyMethod`**— La manière dont vous souhaitez que les modifications soient appliquées à ce paramètre. Les valeurs autorisées sont `immediate` et `pending-reboot`.

> **Note**
>
> Les paramètres avec le `ApplyType` de `static` doivent avoir une `ApplyMethod` de `pending-reboot`.

1. Pour modifier les valeurs du paramètre `change_stream_log_retention_duration`, exécutez la commande suivante et remplacez `parameter-value` par la valeur que vous souhaitez attribuer au paramètre.

   Pour Linux, macOS ou Unix :

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group \
```

```
    --parameters
 "ParameterName=change_stream_log_retention_duration,ParameterValue=<parameter-
value>,ApplyMethod=immediate"
```

Pour Windows :

```
aws docdb modify-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name sample-parameter-group ^
    --parameters
 "ParameterName=change_stream_log_retention_duration,ParameterValue=<parameter-
value>,ApplyMethod=immediate"
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "DBClusterParameterGroupName": "sample-parameter-group"
}
```

2. Patientez au moins 5 minutes.
3. Répertoriez les valeurs de paramètre de `sample-parameter-group` pour vous assurer que vos modifications ont été prises en compte.

   Pour Linux, macOS ou Unix :

   ```
   aws docdb describe-db-cluster-parameters \
       --db-cluster-parameter-group-name sample-parameter-group
   ```

   Pour Windows :

   ```
   aws docdb describe-db-cluster-parameters ^
       --db-cluster-parameter-group-name sample-parameter-group
   ```

   La sortie de cette opération ressemble à ceci (format JSON).

   ```
   {
       "Parameters": [
           {
               "ParameterName": "audit_logs",
               "ParameterValue": "disabled",
               "Description": "Enables auditing on cluster.",
   ```

```
            "Source": "system",
            "ApplyType": "dynamic",
            "DataType": "string",
            "AllowedValues": "enabled,disabled",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot"
        },
        {
            "ParameterName": "change_stream_log_retention_duration",
            "ParameterValue": "12345",
            "Description": "Duration of time in seconds that the change stream
 log is retained and can be consumed.",
            "Source": "user",
            "ApplyType": "dynamic",
            "DataType": "integer",
            "AllowedValues": "3600-86400",
            "IsModifiable": true,
            "ApplyMethod": "immediate"
        }
    ]
}
```

> **Note**
>
> La conservation des journaux des flux de modifications ne supprimera pas les journaux plus anciens que ceux configurés`change_stream_log_retention_duration`valeur jusqu'à ce que la taille du journal soit supérieure à (>) 51 200 Mo.

En utilisantAWS Lambdaavec Change Streams

Amazon DocumentDB est intégré àAWS Lambdavous pouvez donc utiliser les fonctions Lambda pour traiter les enregistrements d'un flux de modifications. Le mappage des sources d'événements Lambda est une ressource qui peut être utilisée pour appeler des fonctions Lambda afin de traiter les événements Amazon DocumentDB qui n'invoquent pas directement Lambda. Avec Amazon DocumentDB change stream comme source d'événements, vous pouvez créer des applications pilotées par les événements qui répondent aux modifications de vos données. Par exemple, vous pouvez utiliser les fonctions Lambda pour traiter de nouveaux documents, suivre les mises à jour de documents existants ou enregistrer les documents supprimés.

Vous pouvez configurer un mappage de source d'événements pour envoyer des enregistrements depuis votre flux de modifications Amazon DocumentDB vers une fonction Lambda. Les événements peuvent être envoyés un par un ou groupés pour une meilleure efficacité et seront traités dans l'ordre. Vous pouvez configurer le comportement de traitement par lots de votre mappage de sources d'événements en fonction d'une durée de fenêtre temporelle spécifique (0 à 300 secondes) ou d'un nombre d'enregistrements par lots (limite maximale de 10 000 enregistrements). Vous pouvez créer plusieurs mappages de sources d'événements pour traiter les mêmes données avec plusieurs fonctions Lambda, ou pour traiter des éléments distincts provenant de plusieurs flux avec une seule fonction.

Si votre fonction renvoie une erreur, Lambda réessaie le lot jusqu'à ce qu'il soit traité correctement. Si les événements du flux de modifications ont expiré, Lambda désactivera le mappage des sources d'événements. Dans ce cas, vous pouvez créer un nouveau mappage des sources d'événements et le configurer avec la position de départ de votre choix. Les mappages de sources d'événements Lambda traitent les événements au moins une fois en raison de la nature distribuée de ses pollers. Par conséquent, votre fonction Lambda peut recevoir des événements dupliqués dans de rares situations. Suivez les meilleures pratiques pour travailler avecAWS Lambdafonctions et créez des fonctions idempotentes pour éviter les problèmes liés à la duplication d'événements. Pour plus d'informations, voirEn utilisantAWS Lambda consoleavec Amazon DocumentDBdans leAWS LambdaGuide du développeur.

Afin de respecter les bonnes pratiques en matière de performances, la fonction Lambda doit être de courte durée. Pour éviter d'introduire des retards de traitement inutiles, elle ne doit pas non plus exécuter de logique complexe. Pour un flux à haute vitesse en particulier, il est préférable de déclencher des flux de travail asynchrones avec des fonctions de post-traitement par étapes plutôt que des Lambdas synchrones de longue durée. Pour plus d'informations sur AWS Lambda, consultez le Manuel du développeur AWS Lambda.

Limites

Les limites suivantes sont à prendre en compte lors de l'utilisation d'Amazon DocumentDB etAWS Lambda:

- AWS Lambdaest actuellement pris en charge uniquement sur Amazon DocumentDB 4.0 et 5.0.
- AWS Lambdan'est actuellement pas pris en charge sur les clusters élastiques ou les clusters globaux.

- AWS Lambdala taille de la charge utile ne peut pas dépasser 6 Mo. Pour plus d'informations sur les tailles de lots Lambda, consultez « Comportement du traitement par lots » dansMappages de sources d'événements Lambdasection duAWS LambdaGuide du développeur.

Utilisation de la validation du schéma JSON

À l'aide de l'opérateur de requête d'`$jsonSchema`évaluation, vous pouvez valider les documents insérés dans vos collections.

Rubriques

- Création et utilisation de la validation du schéma JSON
- Mots clés pris en charge
- bypassDocumentValidation
- Limites

Création et utilisation de la validation du schéma JSON

Création d'une collection avec validation de schéma

Vous pouvez créer une collection avec des règles de `createCollection` fonctionnement et de validation. Ces règles de validation sont appliquées lors des insertions ou des mises à jour de documents Amazon DocumentDB. L'exemple de code suivant montre les règles de validation pour un ensemble d'employés :

```
db.createCollection("employees", {
   "validator": {
      "$jsonSchema": {
         "bsonType": "object",
         "title": "employee validation",
         "required": [ "name", "employeeId"],
         "properties": {
            "name": {
                  "bsonType": "object",
                  "properties": {
                     "firstName": {
                        "bsonType": ["string"]
                     },
                     "lastName": {
```

```
                        "bsonType": ["string"]
                     }
                  },
                  "additionalProperties" : false
            },
            "employeeId": {
               "bsonType": "string",
               "description": "Unique Identifier for employee"
            },
             "salary": {
               "bsonType": "double"
            },
            "age": {
               "bsonType": "number"
            }
         },
         "additionalProperties" : true
      }
   },
   "validationLevel": "strict", "validationAction": "error"
} )
```

Insérer un document valide

L'exemple suivant insère des documents conformes aux règles de validation du schéma ci-dessus :

```
db.employees.insert({"name" : { "firstName" : "Carol" , "lastName" : "Smith"},
 "employeeId": "c720a" , "salary": 1000.0 })
db.employees.insert({ "name" : { "firstName" : "William", "lastName" : "Taylor" },
 "employeeId" : "c721a", "age" : 24})
```

Insérer un document non valide

L'exemple suivant insère des documents qui ne sont pas conformes aux règles de validation du schéma ci-dessus. Dans cet exemple, la valeur EmployeeID n'est pas une chaîne :

```
db.employees.insert({
    "name" : { "firstName" : "Carol" , "lastName" : "Smith"},
    "employeeId": 720 ,
    "salary": 1000.0
})
```

Cet exemple montre une syntaxe incorrecte dans le document.

Modifier une collection

La `collMod` commande est utilisée pour ajouter ou modifier les règles de validation d'une collection existante. L'exemple suivant ajoute un champ de salaire à la liste des champs obligatoires :

```
db.runCommand({"collMod" : "employees",
   "validator": {
      "$jsonSchema": {
         "bsonType": "object",
         "title": "employee validation",
         "required": [ "name", "employeeId", "salary"],
         "properties": {
            "name": {
                  "bsonType": "object",
                  "properties": {
                     "firstName": {
                        "bsonType": ["string"]
                     },
                     "lastName": {
                        "bsonType": ["string"]
                     }
                  },
                  "additionalProperties" : false
            },
            "employeeId": {
               "bsonType": "string",
               "description": "Unique Identifier for employee"
            },
             "salary": {
               "bsonType": "double"
            },
            "age": {
               "bsonType": "number"
            }
         },
         "additionalProperties" : true
      }
   }
} )
```

Adressage des documents ajoutés avant la modification des règles de validation

Pour traiter les documents qui ont été ajoutés à votre collection avant que les règles de validation ne soient modifiées, utilisez les `validationLevel` modificateurs suivants :

- strict : applique les règles de validation à toutes les insertions et mises à jour.
- modéré : applique les règles de validation aux documents valides existants. Lors des mises à jour, les documents non valides existants ne sont pas vérifiés.

Dans l'exemple suivant, après avoir mis à jour les règles de validation sur la collection nommée « employés », le champ salaire est obligatoire. La mise à jour du document suivant échouera :

```
db.runCommand({
    update: "employees",
    updates: [{
        q: { "employeeId": "c721a" },
        u: { age: 25 , salary : 1000},
        upsert: true }]
})
```

Amazon DocumentDB renvoie le résultat suivant :

```
{
"n" : 0,
    "nModified" : 0,
    "writeErrors" : [
        {
"index" : 0,
            "code" : 121,
            "errmsg" : "Document failed validation"
        }
    ],
    "ok" : 1,
    "operationTime" : Timestamp(1234567890, 1)
}
```

La mise à jour du niveau de validation `moderate` permettra au document ci-dessus d'être correctement mis à jour :

```
db.runCommand({
    "collMod" : "employees",
```

```
    validationLevel : "moderate"
})

db.runCommand({
    update: "employees",
    updates: [{
        q: { "employeeId": "c721a" },
        u: { age: 25 , salary : 1000},
        upsert: true }]
})
```

Amazon DocumentDB renvoie le résultat suivant :

```
{
"n" : 1,
    "nModified" : 1,
    "ok" : 1,
    "operationTime" : Timestamp(1234567890, 1)
}
```

Récupération de documents avec le $JSONSchema

L'`$jsonSchema`opérateur peut être utilisé comme filtre pour interroger les documents qui correspondent au schéma JSON. Il s'agit d'un opérateur de niveau supérieur qui peut être présent dans les documents filtrés sous forme de champ de niveau supérieur ou utilisé avec des opérateurs de requête tels que `$and` `$or`, et `$nor`. Les exemples suivants montrent l'utilisation de $JSONSchema en tant que filtre individuel et avec d'autres opérateurs de filtre :

Document inséré dans une collection « d'employés » :

```
{ "name" : { "firstName" : "Carol", "lastName" : "Smith" }, "employeeId" : "c720a",
 "salary" : 1000 }
{ "name" : { "firstName" : "Emily", "lastName" : "Brown" }, "employeeId" : "c720b",
 "age" : 25, "salary" : 1050.2 }
{ "name" : { "firstName" : "William", "lastName" : "Taylor" }, "employeeId" : "c721a",
 "age" : 24, "salary" : 1400.5 }
{ "name" : { "firstName" : "Jane", "lastName" : "Doe" }, "employeeId" : "c721a",
 "salary" : 1300 }
```

Collection filtrée avec l'`$jsonSchema`opérateur uniquement :

```
db.employees.find({
```

```
       $jsonSchema: { required: ["age"] } })
```

Amazon DocumentDB renvoie le résultat suivant :

```
{ "_id" : ObjectId("64e5f91c6218c620cf0e8f8b"), "name" : { "firstName" : "Emily",
 "lastName" : "Brown" }, "employeeId" : "c720b", "age" : 25, "salary" : 1050.2 }
{ "_id" : ObjectId("64e5f94e6218c620cf0e8f8c"), "name" : { "firstName" : "William",
 "lastName" : "Taylor" }, "employeeId" : "c721a", "age" : 24, "salary" : 1400.5 }
```

Collection filtrée avec l'`$jsonSchema`opérateur et un autre opérateur :

```
db.employees.find({
       $or: [{ $jsonSchema: { required: ["age", "name"]}},
            { salary: { $lte:1000}}]});
```

Amazon DocumentDB renvoie le résultat suivant :

```
{ "_id" : ObjectId("64e5f8886218c620cf0e8f8a"), "name" : { "firstName" : "Carol",
 "lastName" : "Smith" }, "employeeId" : "c720a", "salary" : 1000 }
{ "_id" : ObjectId("64e5f91c6218c620cf0e8f8b"), "name" : { "firstName" : "Emily",
 "lastName" : "Brown" }, "employeeId" : "c720b", "age" : 25, "salary" : 1050.2 }
{ "_id" : ObjectId("64e5f94e6218c620cf0e8f8c"), "name" : { "firstName" : "William",
 "lastName" : "Taylor" }, "employeeId" : "c721a", "age" : 24, "salary" : 1400.5 }
```

Collection filtrée avec l'`$jsonSchema`opérateur et avec `$match` dans le filtre agrégé :

```
db.employees.aggregate(
    [{ $match: {
        $jsonSchema: {
            required: ["name", "employeeId"],
            properties: {"salary" :{"bsonType": "double"}}
        }
       }
    }]
)
```

Amazon DocumentDB renvoie le résultat suivant :

```
{
"_id" : ObjectId("64e5f8886218c620cf0e8f8a"),
 "name" : { "firstName" : "Carol", "lastName" : "Smith" },
```

```
"employeeId" : "c720a",
"salary" : 1000
}
{
"_id" : ObjectId("64e5f91c6218c620cf0e8f8b"),
"name" : { "firstName" : "Emily", "lastName" : "Brown" },
"employeeId" : "c720b",
"age" : 25,
"salary" : 1050.2
}
{
"_id" : ObjectId("64e5f94e6218c620cf0e8f8c"),
"name" : { "firstName" : "William", "lastName" : "Taylor" },
"employeeId" : "c721a",
"age" : 24,
"salary" : 1400.5
}
{
"_id" : ObjectId("64e5f9786218c620cf0e8f8d"),
"name" : { "firstName" : "Jane", "lastName" : "Doe" },
"employeeId" : "c721a",
"salary" : 1300
}
```

Afficher les règles de validation existantes

Pour consulter les règles de validation existantes sur une collection, utilisez :

```
db.runCommand({
    listCollections: 1,
    filter: { name: 'employees' }
})
```

Amazon DocumentDB renvoie le résultat suivant :

```
{
    "waitedMS" : NumberLong(0),
    "cursor" : {
        "firstBatch" : [
            {
                "name" : "employees",
                "type" : "collection",
                "options" : {
```

```
                    "autoIndexId" : true,
                    "capped" : false,
                    "validator" : {
                        "$jsonSchema" : {
                            "bsonType" : "object",
                            "title" : "employee validation",
                            "required" : [
                                "name",
                                "employeeId",
                                "salary"
                            ],
                            "properties" : {
                                "name" : {
                                    "bsonType" : "object",
                                    "properties" : {
                                        "firstName" : {
                                            "bsonType" : [
                                                "string"
                                            ]
                                        },
                                        "lastName" : {
                                            "bsonType" : [
                                                "string"
                                            ]
                                        }
                                    },
                                    "additionalProperties" : false
                                },
                                "employeeId" : {
                                    "bsonType" : "string",
                                    "description" : "Unique Identifier for employee"
                                },
                                "salary" : {
                                    "bsonType" : "double"
                                },
                                "age" : {
                                    "bsonType" : "number"
                                }
                            },
                            "additionalProperties" : true
                        }
                    },
                    "validationLevel" : "moderate",
                    "validationAction" : "error"
```

```
                },
                "info" : {
                    "readOnly" : false
                },
                "idIndex" : {
                    "v" : 2,
                    "key" : {
                        "_id" : 1
                    },
                    "name" : "_id_",
                    "ns" : "test.employees"
                }
            }
        ],
        "id" : NumberLong(0),
        "ns" : "test.$cmd.listCollections"
    },
    "ok" : 1,
    "operationTime" : Timestamp(1692788937, 1)
}
```

Amazon DocumentDB conserve également les règles de validation lors de la phase d'`$out`agrégation.

Mots clés pris en charge

Les champs suivants sont pris en charge dans les `collMod` commandes `create` et :

- **`Validator`**— Supporte `$jsonSchem` l'opérateur a.
- **`ValidationLevel`**— Supports `off` et `moderate` valeurs. `strict`
- **`ValidationAction`**— Supporte la `error` valeur.

L'opérateur $JSONSchema prend en charge les mots clés suivants :

- `additionalItems`
- `additionalProperties`
- `allOf`
- `anyOf`
- `bsonType`

- dependencies
- description
- enum
- exclusiveMaximum
- exclusiveMinimum
- items
- maximum
- minimum
- maxItems
- minItems
- maxLength
- minLength
- maxProperties
- minProperties
- multipleOf
- not
- oneOf
- pattern
- patternProperties
- properties
- required
- title
- type
- uniqueItems

bypassDocumentValidation

Amazon DocumentDB prend en charge bypassDocumentValidation les commandes et méthodes suivantes :

- insert

- `update`
- `findAndModify`
- `$out`étape dans le `aggregate` commandement et dans la `db.collection.aggregate()` méthode

Amazon DocumentDB ne prend pas en charge les commandes suivantes pour : `bypassDocumentValidation`

- `$merge`dans la `aggregate` commande et dans la `db.collection.aggregate()` méthode
- `mapReduce`commande et `db.collection.mapReduce()` méthode
- `applyOps` commande

Limites

Les limites suivantes s'appliquent à la `$jsonSchema` validation :

- Amazon DocumentDB renvoie l'erreur « Échec de la validation du document » lorsqu'une opération ne respecte pas la règle de validation.
- Les clusters élastiques Amazon DocumentDB ne sont pas pris en charge. `$jsonSchema`

Connexion à Amazon DocumentDB en tant qu'ensemble de réplicas

Lorsque vous développez sur Amazon DocumentDB (avec compatibilité MongoDB), nous vous recommandons de vous connecter à votre cluster en tant qu'ensemble de réplicas et de distribuer les lectures aux instances de réplica à l'aide des capacités de préférence de lecture intégrées de votre pilote. Cette section approfondit ce que cela signifie et décrit comment vous connecter à votre cluster Amazon DocumentDB en tant qu'ensemble de réplicas avec comme exemple le SDK pour Python.

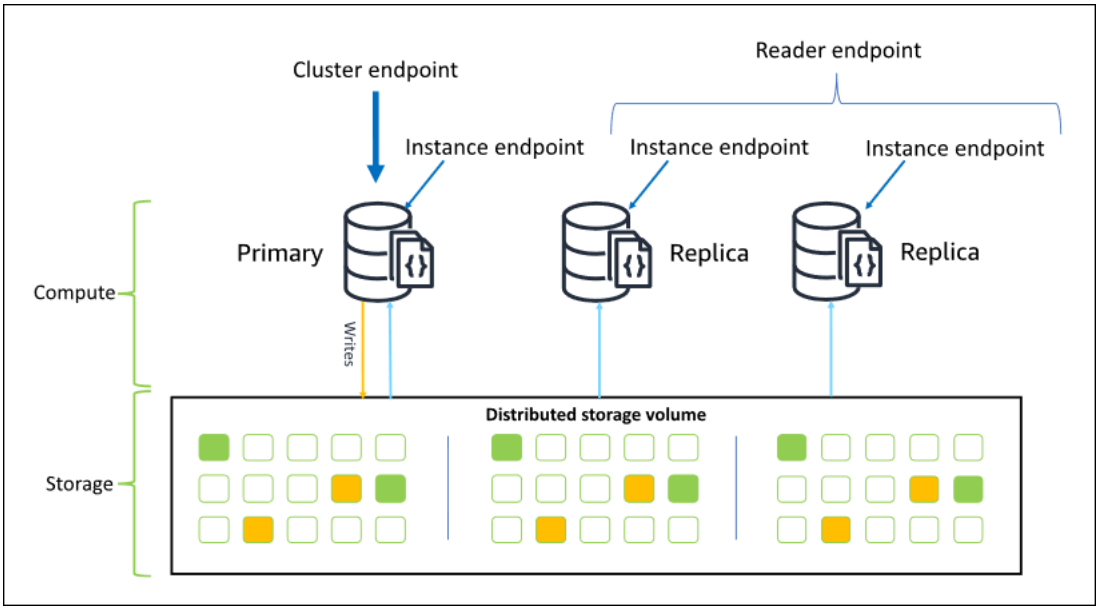
Amazon DocumentDB possède trois points de terminaison que vous pouvez utiliser pour vous connecter à votre cluster :

- Point de terminaison de cluster
- Point de terminaison du lecteur
- Point de terminaison d'instance

Dans la plupart des cas, lorsque vous vous connectez à Amazon DocumentDB, nous vous recommandons d'utiliser le point de terminaison du cluster. Il s'agit d'un CNAME qui pointe vers l'instance principale de votre cluster, comme illustré dans le schéma suivant.

Lorsque vous utilisez un tunnel SSH, nous vous recommandons de vous connecter à votre cluster à l'aide du point de terminaison de cluster et de ne pas essayer de vous connecter en mode d'ensemble de réplicas (c'est-à-dire en spécifiant `replicaSet=rs0` dans votre chaîne de connexion), car cela entraînerait une erreur.

> **Note**
>
> Pour plus d'informations sur les points de terminaison Amazon DocumentDB, consultez.Points de terminaison Amazon DocumentDB.



À l'aide du point de terminaison du cluster, vous pouvez vous connecter à votre cluster en mode jeu de réplicas. Vous pouvez ensuite utiliser les fonctionnalités intégrées du pilote de préférence de lecture. Dans l'exemple suivant, la spécification `/?replicaSet=rs0` signifie au kit SDK que vous souhaitez vous connecter en tant que jeu de réplicas. Si vous omettez `/?replicaSet=rs0'`, le client achemine toutes les demandes vers le point de terminaison du cluster, c'est-à-dire votre instance principale.

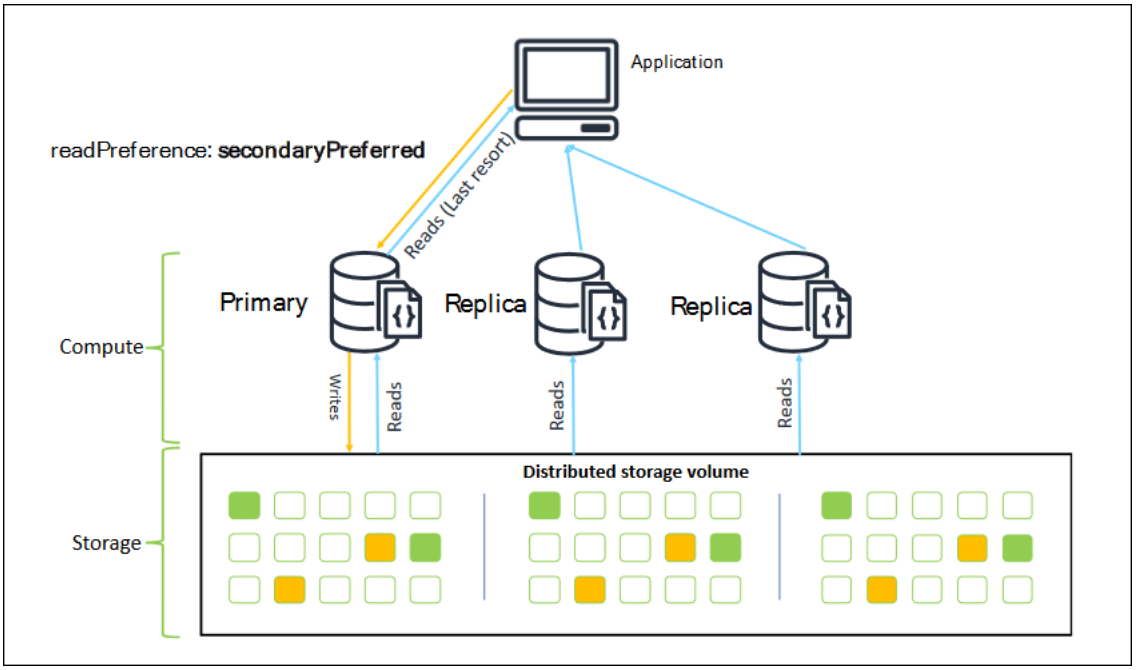
```
## Create a MongoDB client, open a connection to Amazon DocumentDB as a
   ## replica set and specify the read preference as secondary preferred
```

```
   client = pymongo.MongoClient('mongodb://<user-name>:<password>@mycluster.node.us-
east-1.docdb.amazonaws.com:27017/?replicaSet=rs0')
```

L'avantage de se connecter en tant qu'ensemble de réplicas est qu'il permet à votre kit SDK de découvrir automatiquement la topographie du cluster, y compris lorsque des instances sont ajoutées ou supprimées du cluster. Vous pouvez ensuite utiliser votre cluster plus efficacement en acheminant les demandes de lecture vers vos instances de réplica.

Lorsque vous vous connectez en tant que jeu de réplicas, vous pouvez spécifier la `readPreference` pour la connexion. Si vous spécifiez comme préférence de lecture `secondaryPreferred`, le client achemine les requêtes en lecture vers vos réplicas et écrit les requêtes vers votre instance principale (comme dans le diagramme suivant). Il s'agit d'une meilleure utilisation des ressources de votre cluster. Pour plus d'informations, consultez Options de préférence de lecture.

```
## Create a MongoDB client, open a connection to Amazon DocumentDB as a
   ##   replica set and specify the read preference as secondary preferred
client = pymongo.MongoClient('mongodb://<user-name>:<password>@mycluster.node.us-
east-1.docdb.amazonaws.com:27017/?replicaSet=rs0&readPreference=secondaryPreferred')
```



Les lectures des réplicas Amazon DocumentDB sont cohérentes à terme. Elles renvoient les données dans le même ordre qu'elles ont été écrites sur le nœud principal, et le retard de réplication est souvent inférieur à 50 ms. Vous pouvez surveiller le retard de réplica de votre cluster à l'aide des

métriques Amazon CloudWatch.`DBInstanceReplicaLag`et`DBClusterReplicaLagMaximum`. Pour plus d'informations, consultez Surveillance d'Amazon DocumentDB avec CloudWatch.

Contrairement à l'architecture de base de données monolithique traditionnelle, Amazon DocumentDB sépare le stockage et le calcul. En raison de cette architecture moderne, nous vous encourageons à effectuer une mise à l'échelle en lecture sur les instances de réplica. Les lectures sur les instances de réplica ne bloquent pas les écritures répliquées à partir de l'instance principale. Vous pouvez ajouter jusqu'à 15 instances de réplica en lecture dans un cluster et monter en charge pour atteindre plusieurs millions de lectures par seconde.

L'avantage clé de la connexion en tant qu'ensemble de réplicas et de la distribution des lectures sur les réplicas est qu'elle augmente les ressources globales de votre cluster disponibles pour travailler pour votre application. Nous vous recommandons de vous connecter en tant qu'ensemble de réplicas à titre de bonne pratique. De plus, nous le recommandons le plus souvent dans les scénarios suivants :

- Vous utilisez presque 100 % d'UC sur votre instance principale.
- Le taux d'accès au cache du tampon est proche de zéro.
- Vous atteignez les limites de connexion ou de curseur pour une instance individuelle.

La mise à l'échelle d'une taille d'instance de cluster est une option et, dans certains cas, peut être le meilleur moyen de dimensionner le cluster. Mais vous devez également réfléchir à la façon de mieux utiliser les réplicas que vous avez déjà dans votre cluster. Cela vous permet d'augmenter la mise à l'échelle sans avoir à augmenter le coût d'utilisation d'un type d'instance plus grand. Nous vous recommandons également de surveiller et d'alerter ces limites (c'est-à-dire :`CPUUtilization`,`DatabaseConnections`, et`BufferCacheHitRatio`) à l'aide d'alarmes CloudWatch afin de savoir quand une ressource est massivement utilisée.

Pour plus d’informations, consultez les rubriques suivantes :

- Bonnes pratiques pour Amazon DocumentDB
- Quotas et limites Amazon DocumentDB

Utilisation des connexions de cluster

Envisagez le scénario d'utilisation de toutes les connexions de votre cluster. Par exemple, une instance `r5.2xlarge` a une limite de 4 500 connexions (et 450 curseurs ouverts). Si vous créez un

cluster Amazon DocumentDB à trois instances et que vous vous connectez uniquement à l'instance principale à l'aide du point de terminaison du cluster, les limites de votre cluster pour les connexions ouvertes et les curseurs sont de 4 500 et 450 respectivement. Vous pouvez atteindre ces limites si vous construisez des applications qui utilisent de nombreux travaux qui s'exécutent dans des conteneurs. Les conteneurs ouvrent un certain nombre de connexions simultanément et saturent le cluster.

Au lieu de cela, vous pouvez vous connecter au cluster Amazon DocumentDB en tant qu'ensemble de réplicas et distribuer vos lectures entre les instances de réplica. Vous pouvez ensuite tripler efficacement le nombre de connexions et de curseurs disponibles dans le cluster pour atteindre 13 500 et 1 350 respectivement. L'ajout d'instances supplémentaires au cluster augmente uniquement le nombre de connexions et de curseurs pour les charges de travail de lecture. Si vous avez besoin d'augmenter le nombre de connexions pour les écritures dans votre cluster, nous vous recommandons d'augmenter la taille d'instance.

> **Note**
>
> Le nombre de connexions pour les instances `large`, `xlarge` et `2xlarge` augmente avec la taille de l'instance jusqu'à 4 500. Le nombre maximal de connexions par instance pour les instances `4xlarge` ou supérieures est de 4 500. Pour plus d'informations sur les limites par type d'instance, consultez Limites d’instance.

En général, nous vous déconseillons de vous connecter à votre cluster à l'aide de `secondary` comme préférence en lecture. Cela est dû au fait que s'il n'y a pas d'instances de réplica dans votre cluster, les lectures échouent. Par exemple, supposons que vous ayez un cluster Amazon DocumentDB à deux instances avec un nœud principal et un réplica. Si le réplica a un problème, les demandes de lecture à partir d'un groupe de connexions défini comme `secondary` échouent. L'avantage de `secondaryPreferred` est que si le client ne parvient pas à trouver une instance de réplica appropriée à laquelle se connecter, il revient à l'instance principale pour les lectures.

Plusieurs groupes de connexions

Dans certains scénarios, les lectures d'une application doivent avoir une cohérence en lecture après écriture, qui peut être utilisée uniquement à partir de l'instance principale dans Amazon DocumentDB. Dans ces scénarios, vous pouvez créer deux groupes de connexions client : un pour les écritures et un pour les lectures nécessitant une cohérence en lecture après écriture. Pour ce faire, votre code doit ressembler à ce qui suit.

```
## Create a MongoDB client,
##    open a connection to Amazon DocumentDB as a replica set and specify the
 readPreference as primary
clientPrimary = pymongo.MongoClient('mongodb://<user-
name>:<password>@mycluster.node.us-east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0&readPreference=primary')

## Create a MongoDB client,
##    open a connection to Amazon DocumentDB as a replica set and specify the
 readPreference as secondaryPreferred
secondaryPreferred = pymongo.MongoClient('mongodb://<user-
name>:<password>@mycluster.node.us-east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0&readPreference=secondaryPreferred')
```

Une autre option consiste à créer un seul groupe de connexions et à remplacer la préférence de lecture pour une collection donnée.

```
##Specify the collection and set the read preference level for that collection
col = db.review.with_options(read_preference=ReadPreference.SECONDARY_PREFERRED)
```

Récapitulatif

Pour mieux utiliser les ressources de votre cluster, nous vous recommandons de vous connecter à votre cluster à l'aide du mode du jeu de réplicas. Si cela convient à votre application, vous pouvez la dimensionner en répartissant vos lectures sur les instances de réplica.

Connexion à un cluster Amazon DocumentDB depuis l'extérieur d'un Amazon VPC

Les clusters Amazon DocumentDB (en compatibilité MongoDB) sont déployés dans un Amazon Virtual Private Cloud (Amazon VPC). Ils sont accessibles directement par les instances Amazon EC2 ou d'autresAWS services déployés dans le même Amazon VPC. En outre, Amazon DocumentDB est accessible par des instances EC2 ou d'autresAWS services dans différents VPC de la même régionRégion AWS ou d'autres régions via le peering de VPC.

Supposons toutefois que votre cas d'utilisation nécessite que vous (ou votre application) accédiez à vos ressources Amazon DocumentDB depuis l'extérieur du VPC du cluster. Dans ce cas, vous pouvez utiliser le tunneling SSH (également appelé redirection de port) pour accéder à vos ressources Amazon DocumentDB.

L'étude approfondie du tunneling SSH dépasse le cadre de cette rubrique. Pour plus d'information sur le tunneling SSH, consultez la documentation suivante :

- Tunnel SSH
- Exemple de réacheminement de port SSH, en particulier la rubrique Réacheminement local

Pour créer un tunnel SSH, vous devez disposer d'une instance Amazon EC2 s'exécutant dans le même Amazon VPC que le cluster Amazon DocumentDB. Vous pouvez soit utiliser une instance EC2 existante dans le même VPC que votre cluster, soit en créer une. Pour plus d'informations, consultez la rubrique correspondant à votre système d'exploitation :

- Mise en en en en en en Amazon EC2 en en en en
- Mise en en en en en en Amazon EC2 en en en en

Vous pouvez généralement vous connecter à une instance EC2 avec la commande suivante.

```
ssh -i "ec2Access.pem" ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com
```

Si tel est le cas, vous pouvez configurer un tunnel SSH vers le cluster Amazon DocumentDB`sample-cluster.node.us-east-1.docdb.amazonaws.com` en exécutant la commande suivante sur votre ordinateur local. L'indicateur `-L` est utilisé pour réacheminer un port local. Lorsque vous utilisez un tunnel SSH, nous vous recommandons de vous connecter à votre cluster à l'aide du point de terminaison de cluster et de ne pas essayer de vous connecter en mode d'ensemble de réplicas (c'est-à-dire en spécifiant `replicaSet=rs0` dans votre chaîne de connexion), car cela entraînerait une erreur.

```
ssh -i "ec2Access.pem" -L 27017:sample-cluster.node.us-east-1.docdb.amazonaws.com:27017
 ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com -N
```

Une fois le tunnel SSH créé, toutes les commandes que vous envoyez`localhost:27017` sont transmises au cluster Amazon DocumentDB`sample-cluster` s'exécutant dans Amazon VPC. Si le protocole TLS (Transport Layer Security) est activé sur votre cluster Amazon DocumentDB, vous devez télécharger la clé publique d'Amazon DocumentDB à partir dehttps://truststore.pki.rds.amazonaws.com/global/global-bundle.pem. L'opération suivante permet de télécharger ce fichier :

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

> **Note**
>
> Le protocole TLS est activé par défaut pour les nouveaux clusters Amazon DocumentDB. Cependant, vous pouvez la ou le désactiver. Pour plus d'informations, veuillez consulter Gestion des paramètres TLS du cluster Amazon DocumentDB.

Pour vous connecter à votre cluster Amazon DocumentDB depuis l'extérieur d'Amazon VPC, utilisez la commande suivante.

```
mongo --sslAllowInvalidHostnames --ssl --sslCAFile global-bundle.pem --username
 <yourUsername> --password <yourPassword>
```

Connexion à un cluster Amazon DocumentDB depuis Studio 3T

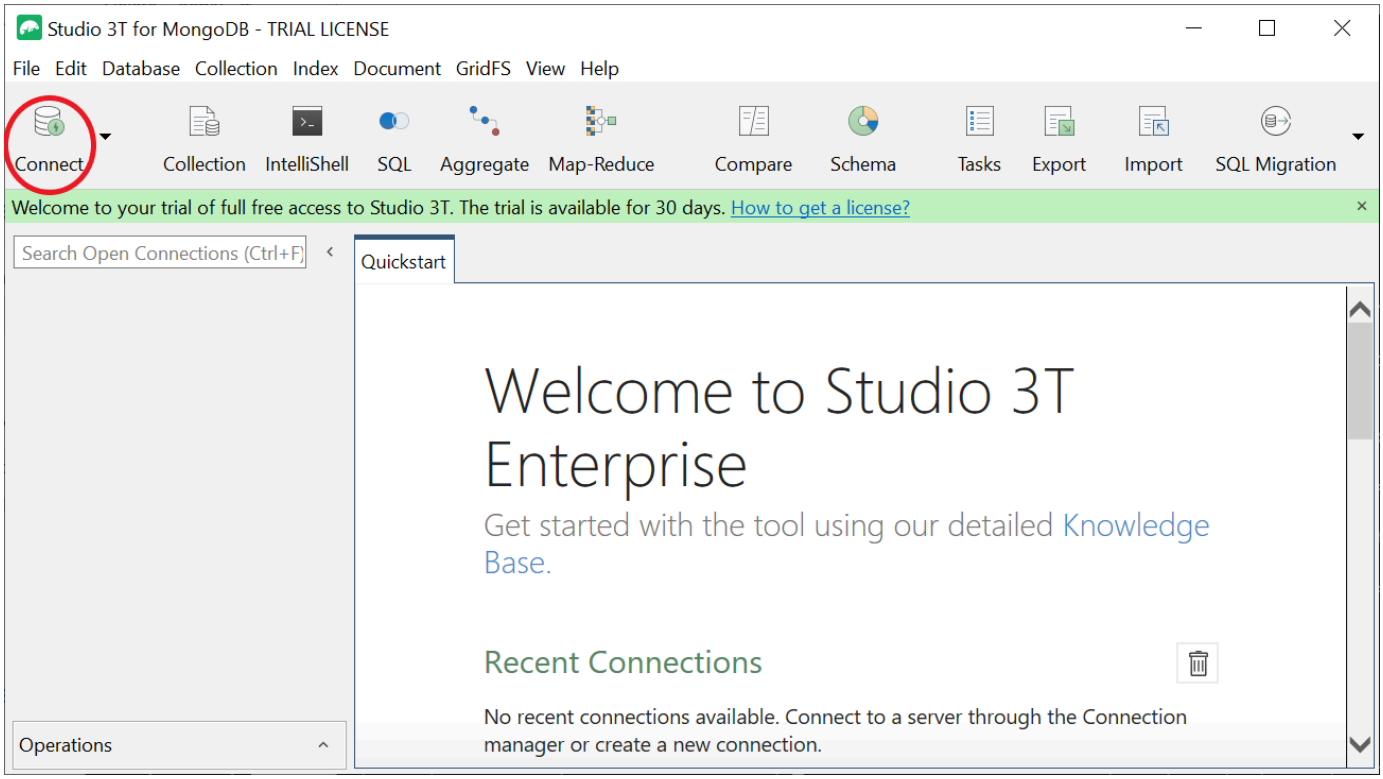
Studio 3T est une interface graphique et un IDE populaires pour les développeurs et les ingénieurs de données qui travaillent avec MongoDB. Il offre plusieurs fonctionnalités puissantes : des vues arborescentes, tabulaires et JSON de vos données, une importation/exportation facile au format CSV, JSON, SQL et BSON/MongoDump, une option de requête flexible, une drag-and-drop interface utilisateur visuelle, un shell mongo intégré avec auto-complétion, un éditeur de pipeline d'agrégation et un support de requêtes SQL.

Prérequis

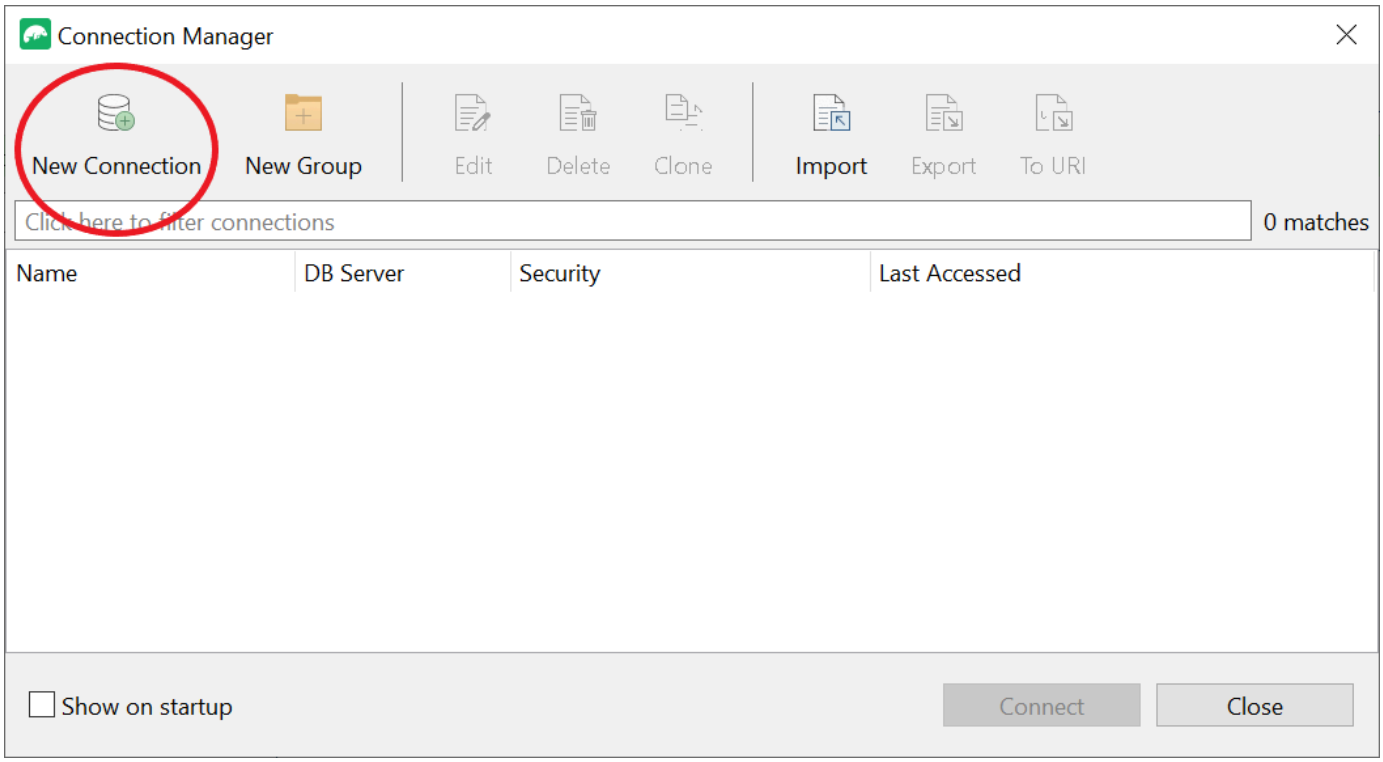
- Si vous ne possédez pas encore de cluster Amazon DocumentDB utilisant Amazon EC2 comme hôte bastion/jump, suivez les instructions pour vous connecter à Amazon EC2.
- Si vous ne possédez pas Studio 3T, téléchargez-le et installez-le.

Connect avec Studio 3T

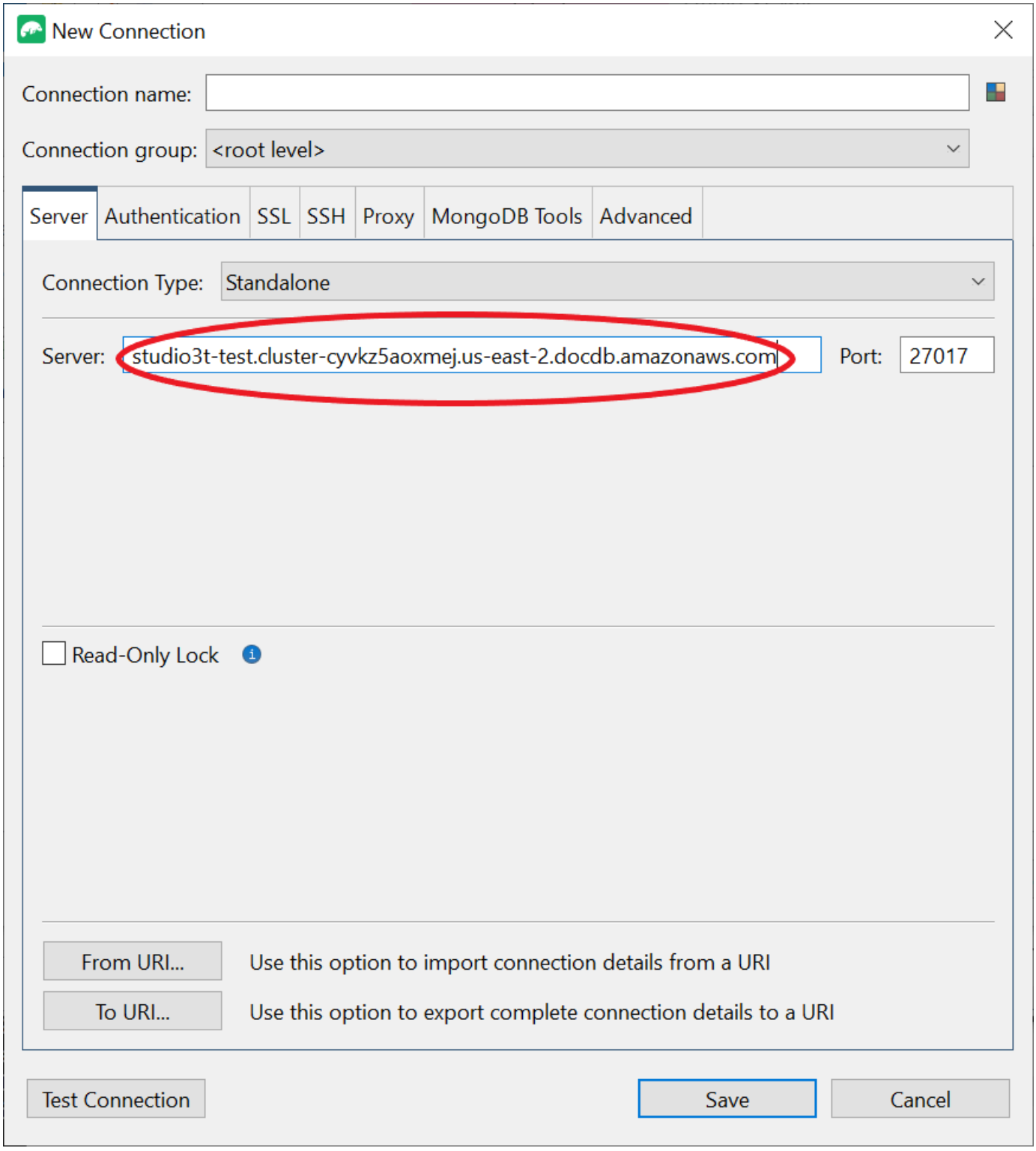
1. Choisissez Connect dans le coin supérieur gauche de la barre d'outils.



2. Choisissez Nouvelle connexion dans le coin supérieur gauche de la barre d'outils.

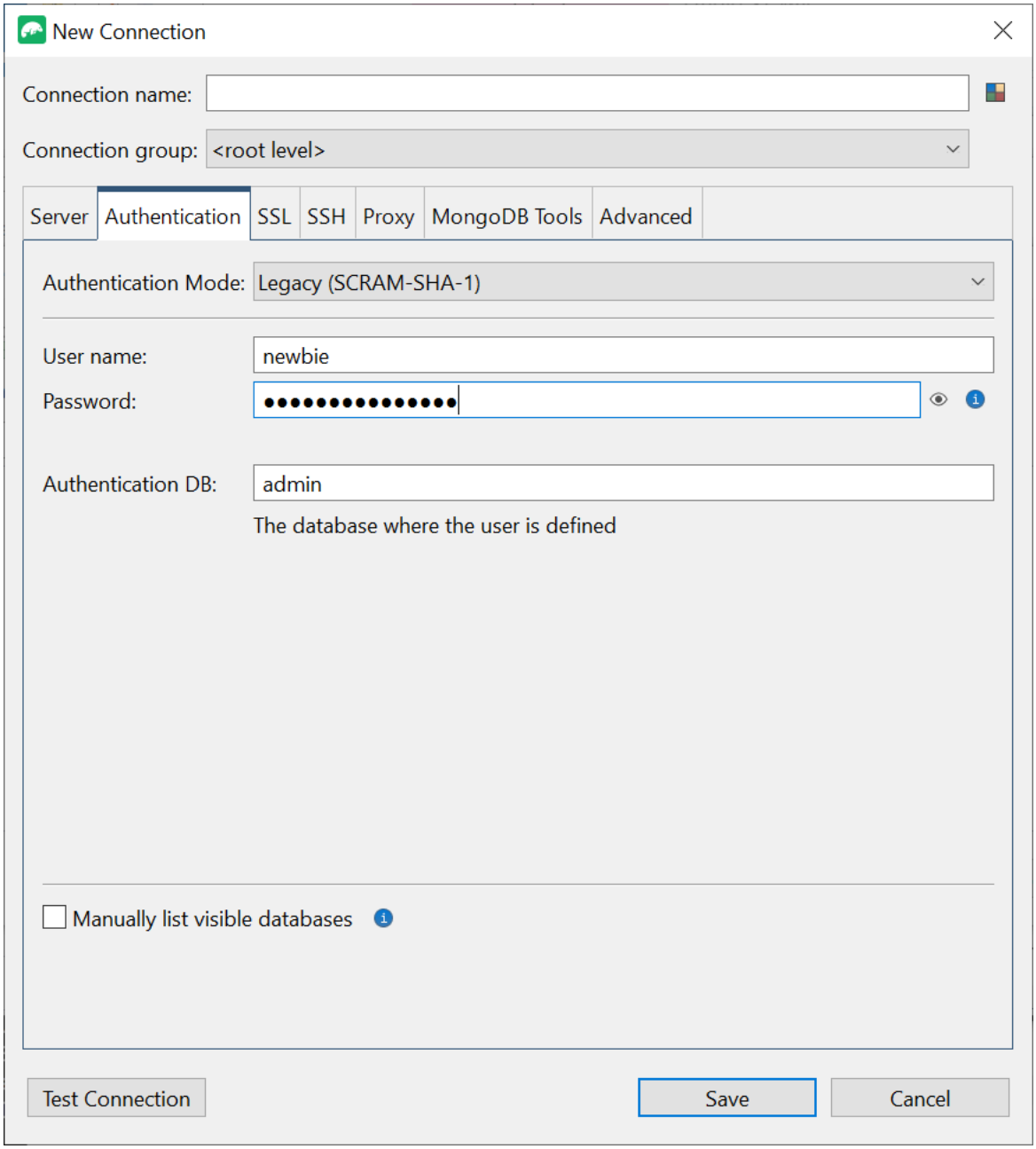


3. Dans l'onglet Serveur, dans le champ Serveur, entrez les informations du point de terminaison du cluster.


New Connection
Connection name:
Connection group: <root level>
Server
Authentication
SSL
SSH
Proxy
MongoDB Tools
Advanced
Connection Type: Standalone
Server: studio3t-test.cluster-cyvkz5aoxmej.us-east-2.docdb.amazonaws.com
Port: 27017
Read-Only Lock
From URI...
Use this option to import connection details from a URI
To URI...
Use this option to export complete connection details to a URI
Test Connection
Save
Cancel

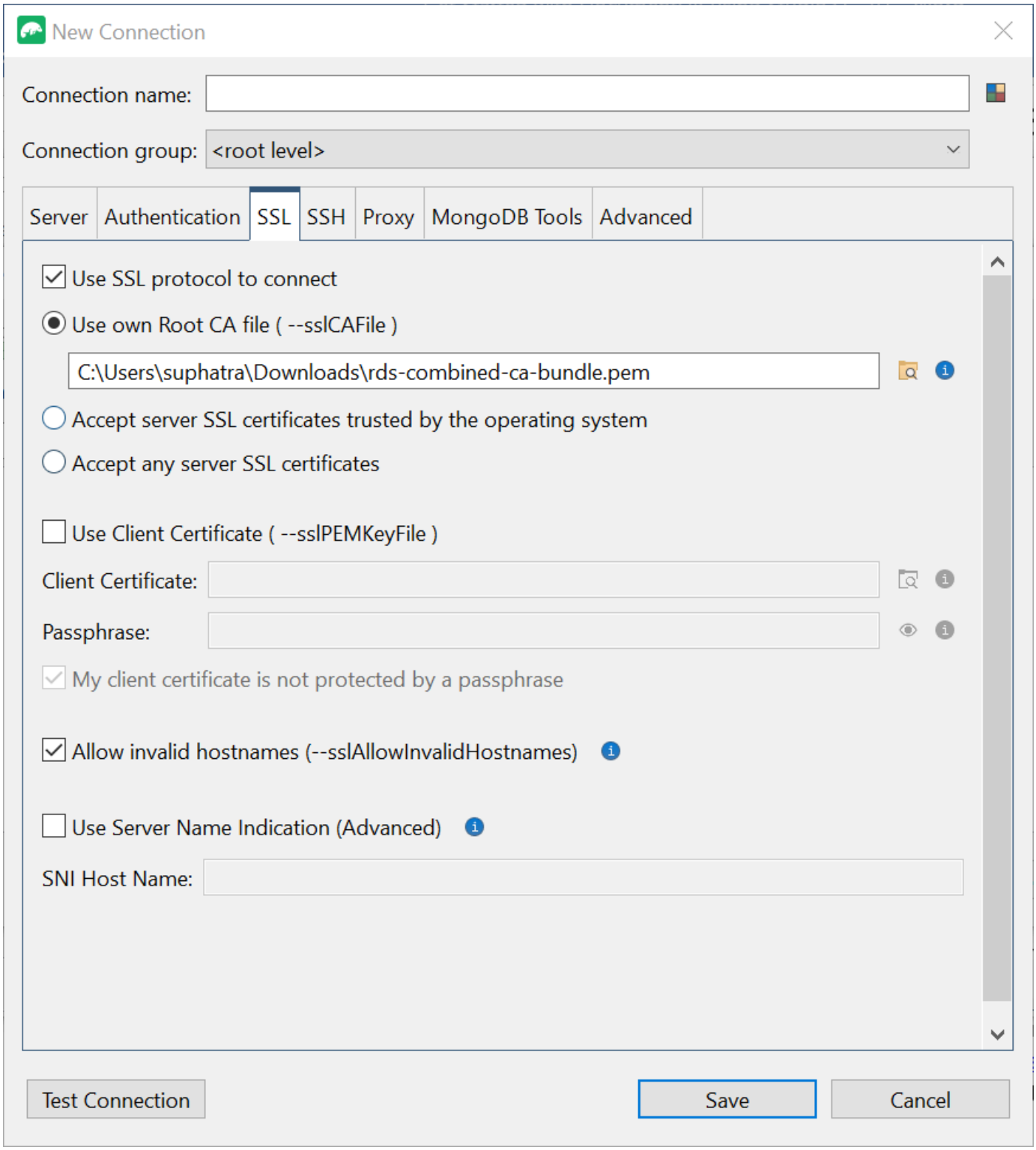
> **Note**
>
> Vous ne trouvez pas le point de terminaison de votre cluster ? Il vous suffit de suivre les étapes indiquées ici.

4. Choisissez l'onglet Authentification et sélectionnez Legacy dans le menu déroulant du mode d'authentification.

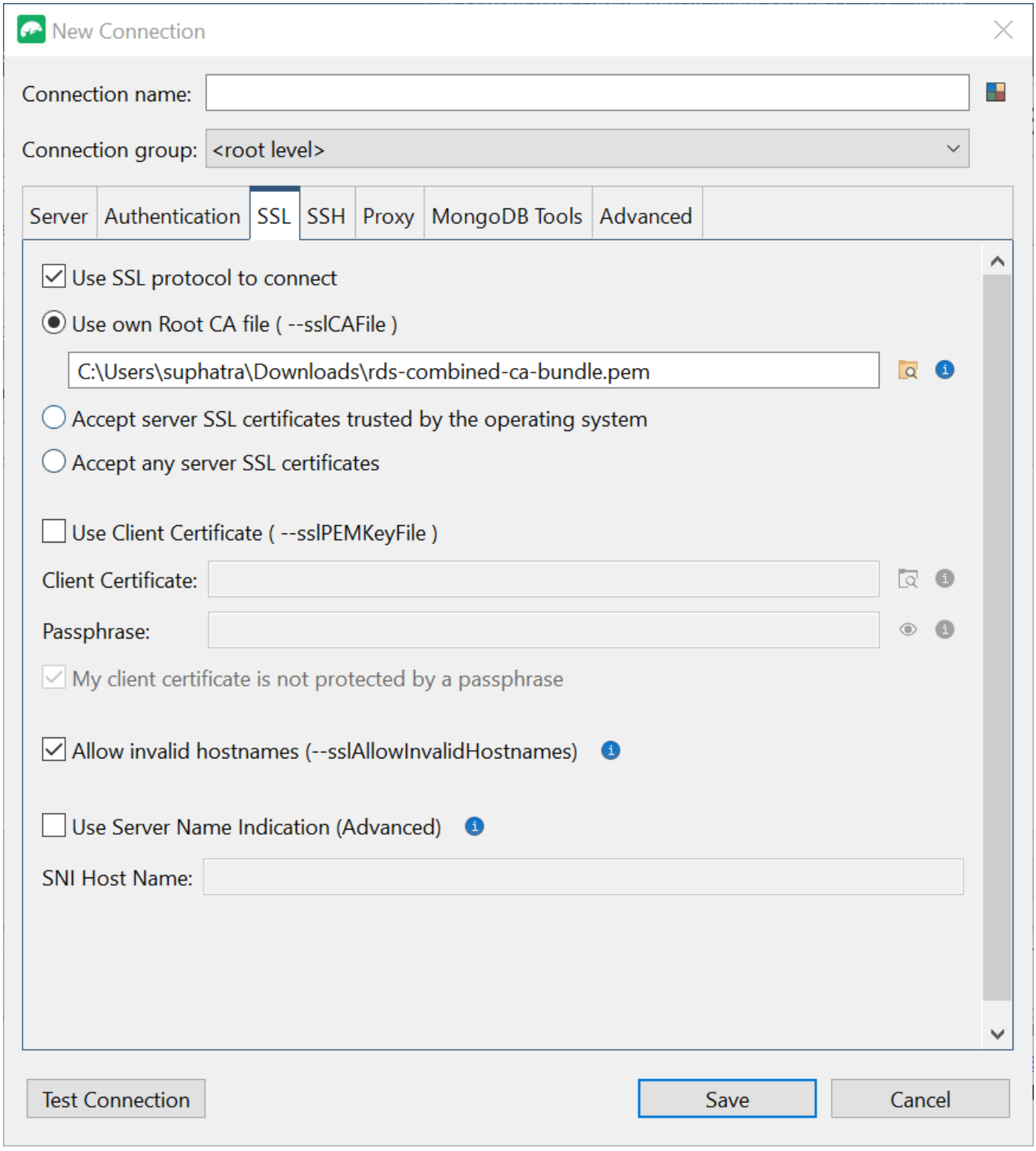


5. Entrez votre nom d'utilisateur et vos informations d'identification dans les champs Nom d'utilisateur et Mot de passe.

6. Choisissez l'onglet SSL et cochez la case Utiliser le protocole SSL pour vous connecter.



7. Choisissez Utiliser votre propre fichier racine CA. Ajoutez ensuite le certificat Amazon DocumentDB (vous pouvez ignorer cette étape si le protocole SSL est désactivé sur votre cluster DocumentDB). Cochez la case pour autoriser les noms d'hôtes non valides.



> **Note**
>
> Vous n'avez pas le certificat ? Vous pouvez le télécharger à l'aide de la commande suivante :
> `wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem`

8. Si vous vous connectez depuis une machine cliente extérieure à Amazon VPC, vous devez créer un tunnel SSH. Vous allez le faire dans l'onglet SSH.

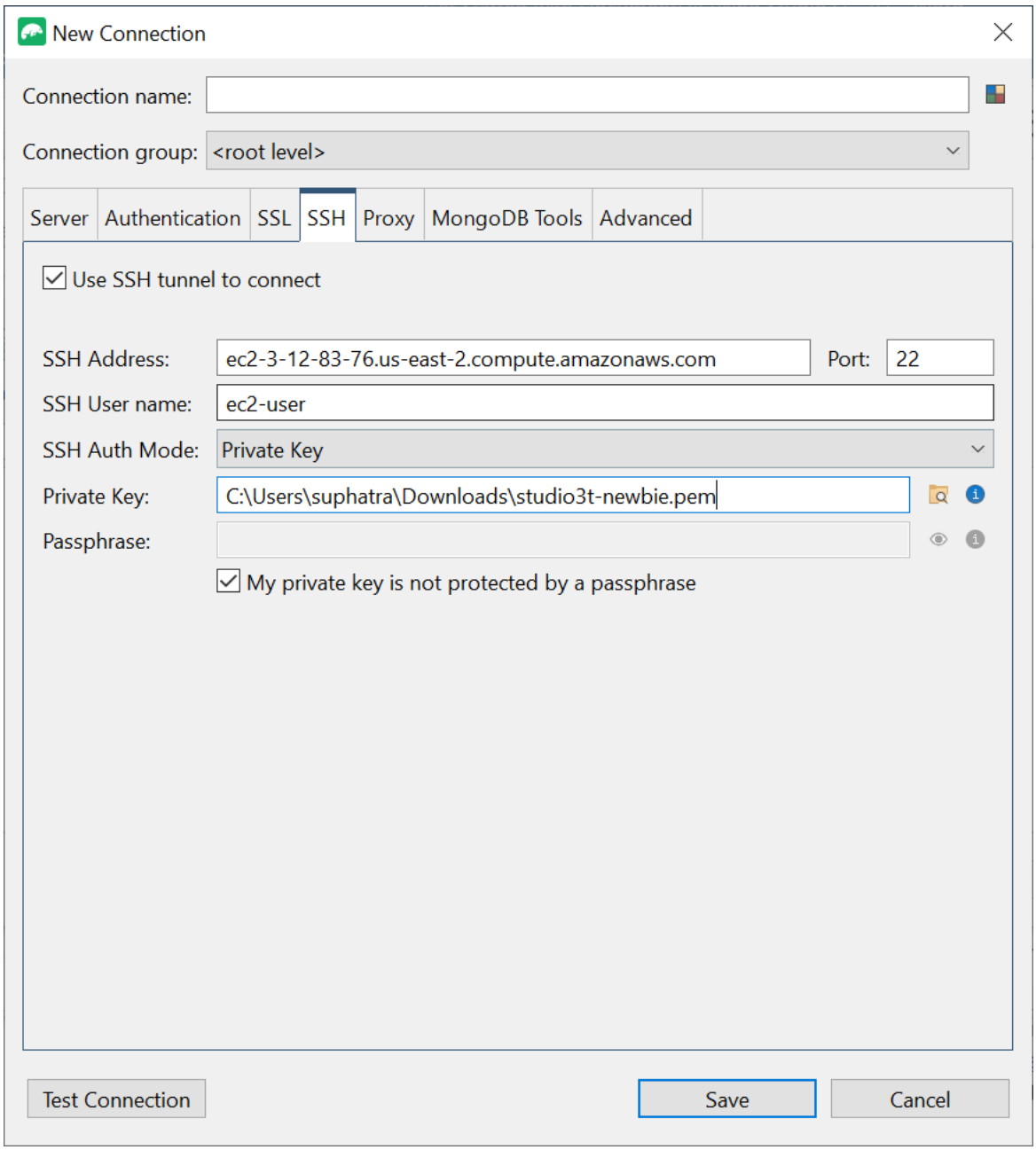
   a. Cochez la case Utiliser le tunnel SSH et saisissez l'adresse SSH dans le champ Adresse SSH. Il s'agit du DNS public (IPV4) de votre instance. Vous pouvez obtenir cette URL depuis votre console de gestion Amazon EC2.

   b. Entrez votre nom d'utilisateur. Il s'agit du nom d'utilisateur de votre instance Amazon EC2

   c. Pour le mode d'authentification SSH, sélectionnez Clé privée. Dans le champ Clé privée, cliquez sur l'icône de recherche de fichiers pour localiser et choisir la clé privée de votre instance Amazon EC2. Il s'agit du fichier .pem (paire de clés) que vous avez enregistré lors de la création de votre instance dans la console Amazon EC2.

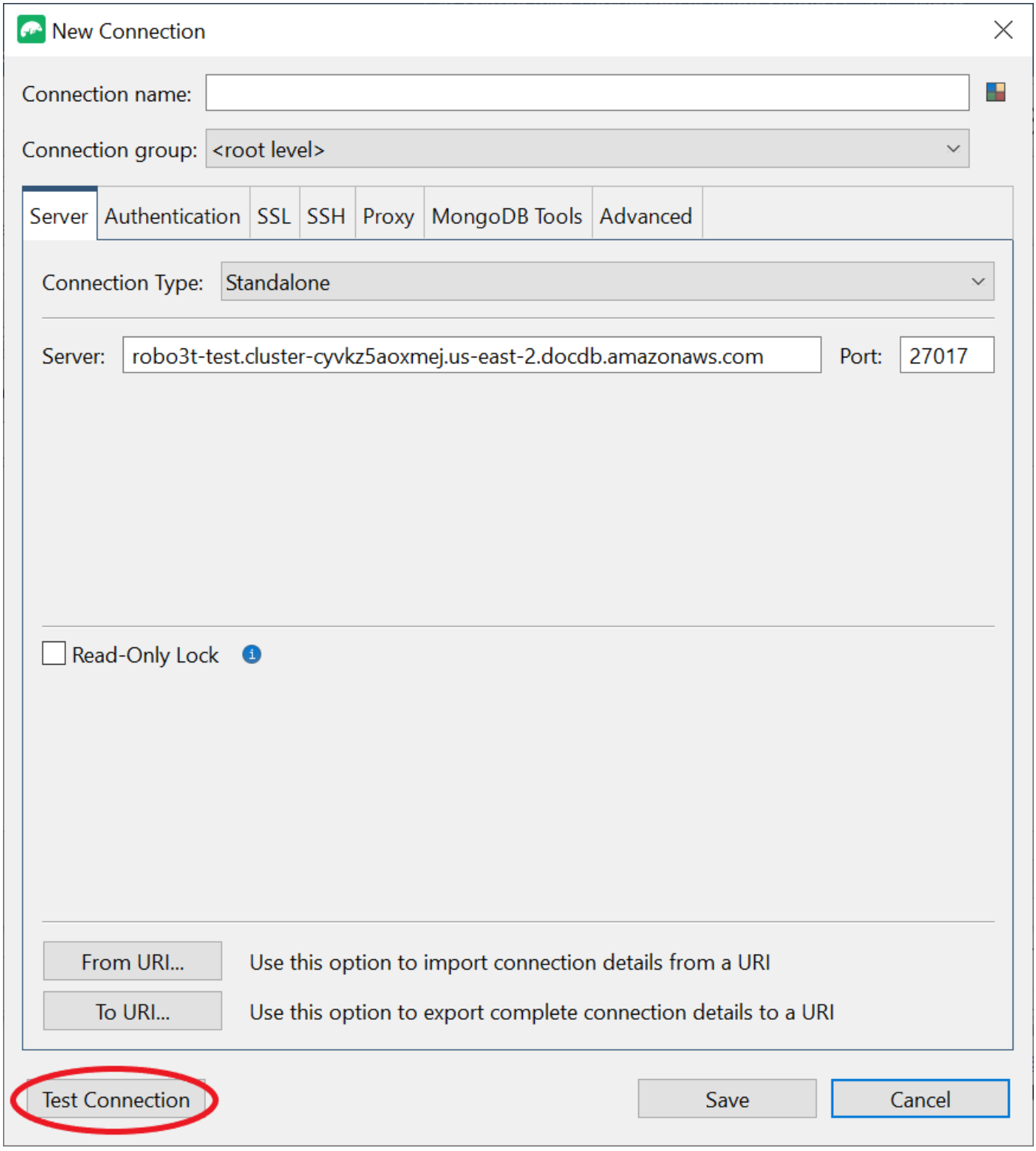
   d. Si vous utilisez un ordinateur client Linux/macOS, vous devrez peut-être modifier les autorisations de votre clé privée à l'aide de la commande suivante :

      ```
      chmod 400 /fullPathToYourPemFile/<yourKey>.pem
      ```

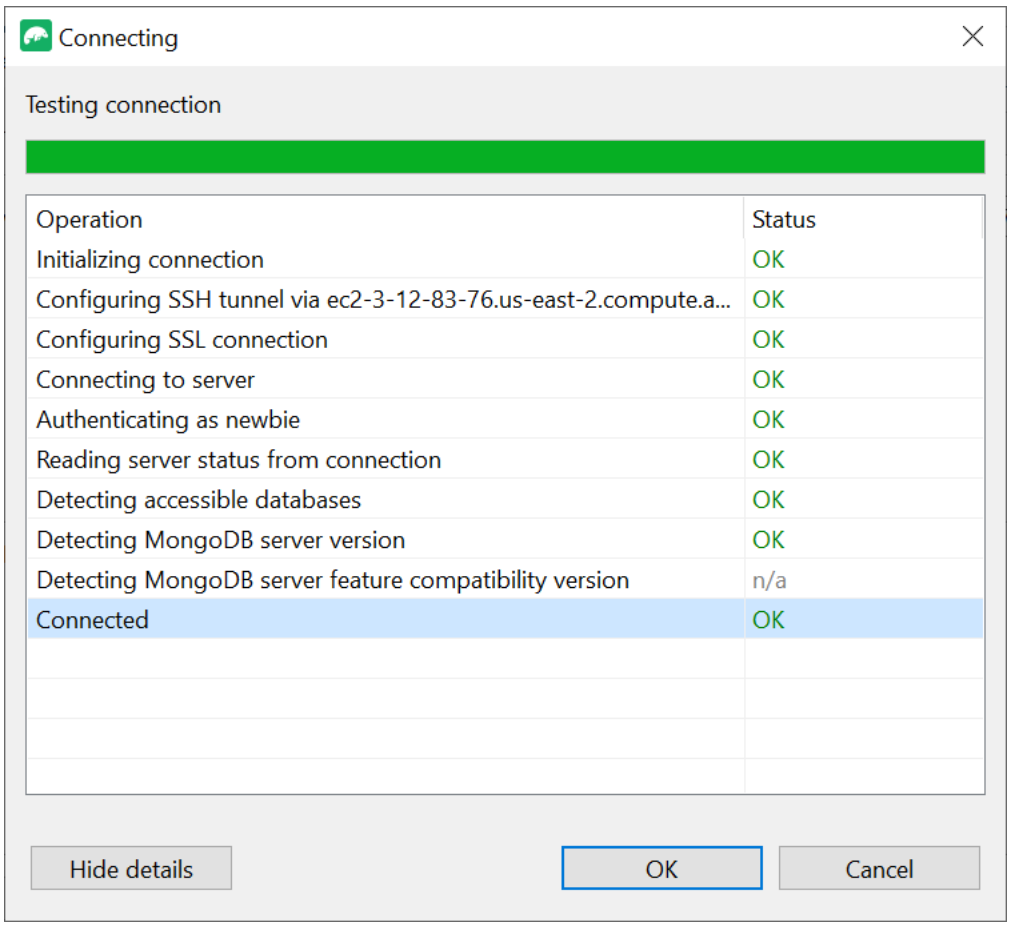


> **Note**
>
> Cette instance Amazon EC2 doit se trouver dans le même VPC Amazon et le même groupe de sécurité que votre cluster DocumentDB. Vous pouvez obtenir l'adresse SSH, le nom d'utilisateur et la clé privée depuis votre console de gestion Amazon EC2.

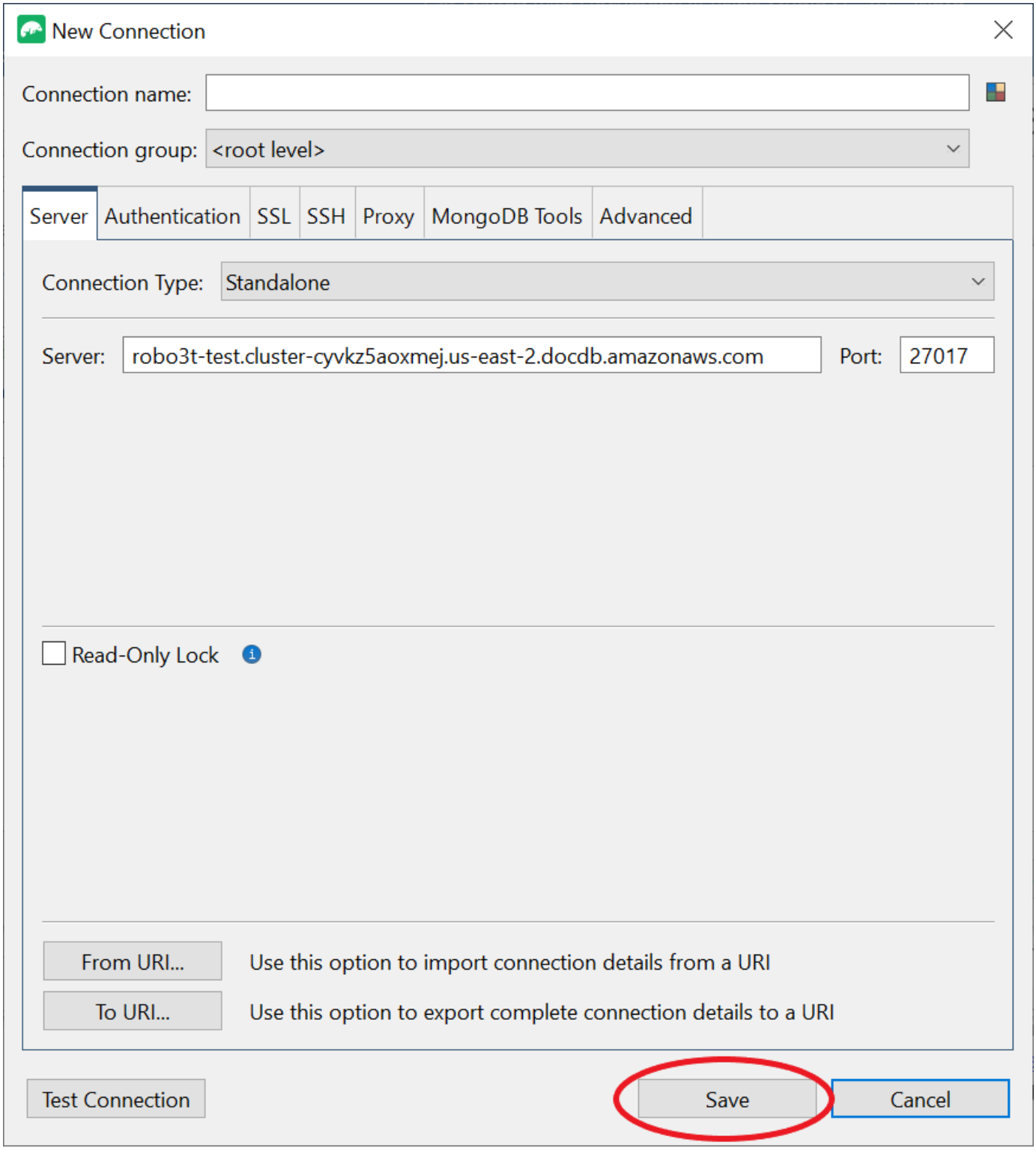
9. Testez maintenant votre configuration en cliquant sur le bouton Tester la connexion.



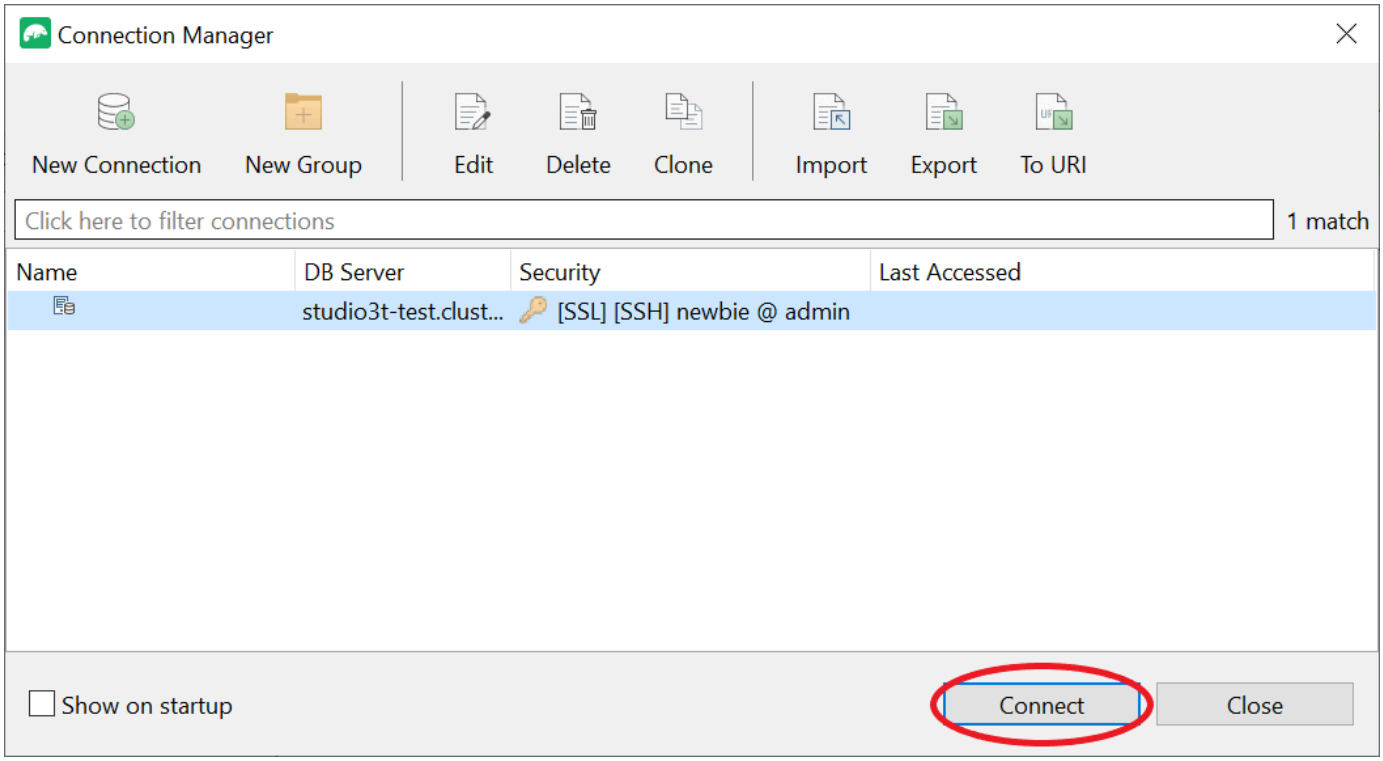
10. Une fenêtre de diagnostic doit charger une barre verte pour indiquer que le test a réussi. Choisissez maintenant OK pour fermer la fenêtre de diagnostic.



11. Choisissez Enregistrer pour enregistrer votre connexion en vue d'une utilisation future.



12. Sélectionnez maintenant votre cluster et choisissez Connect.



Félicitations ! Vous êtes désormais connecté avec succès à votre cluster Amazon DocumentDB via Studio 3T.

Connectez-vous à Amazon DocumentDB à l'aide de DataGrip

DataGripest un puissant environnement de développement intégré (IDE) qui prend en charge différents systèmes de base de données, notamment Amazon DocumentDB. Cette section explique les étapes à suivre pour vous connecter à votre cluster Amazon DocumentDB à l'aide d'une interface graphique DataGrip, ce qui vous permet de gérer et d'interroger facilement vos données.

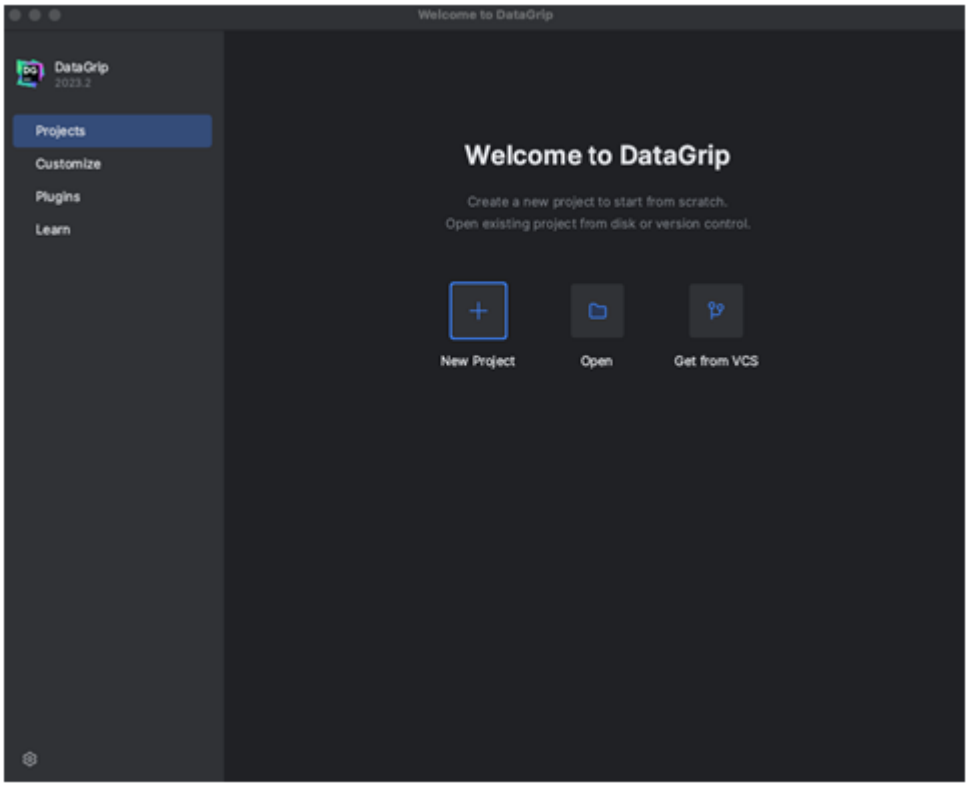
Prérequis

- DataGrip IDE installé sur votre machine. Vous pouvez le télécharger depuis JetBrains.
- Une instance Amazon EC2 exécutée dans le même VPC que votre cluster Amazon DocumentDB. Vous allez utiliser cette instance pour établir un tunnel sécurisé entre votre machine locale et Amazon DocumentDBCluster. Suivez les instructions pour savoir comment procéderConnectez-vous à l'aide d'Amazon EC2.

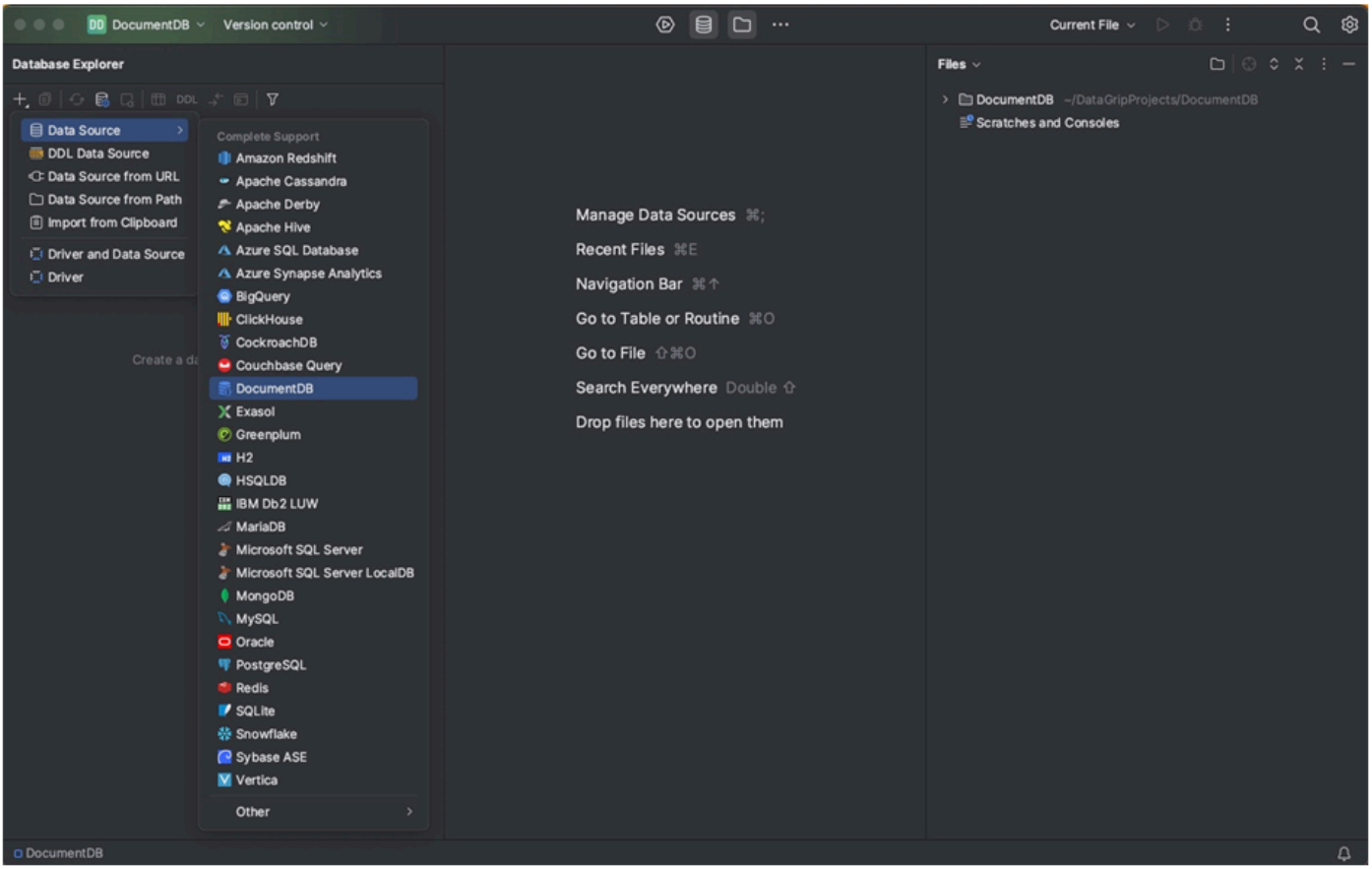
- Alternative à une instance Amazon EC2, à une connexion VPN ou si vous accédez déjà à votre AWS infrastructure à l'aide d'un VPN sécurisé. Si vous préférez cette option, suivez les instructions pour accéder en toute sécurité à Amazon DocumentDB à l'aide de. AWS Client VPN

Connect en utilisant DataGrip

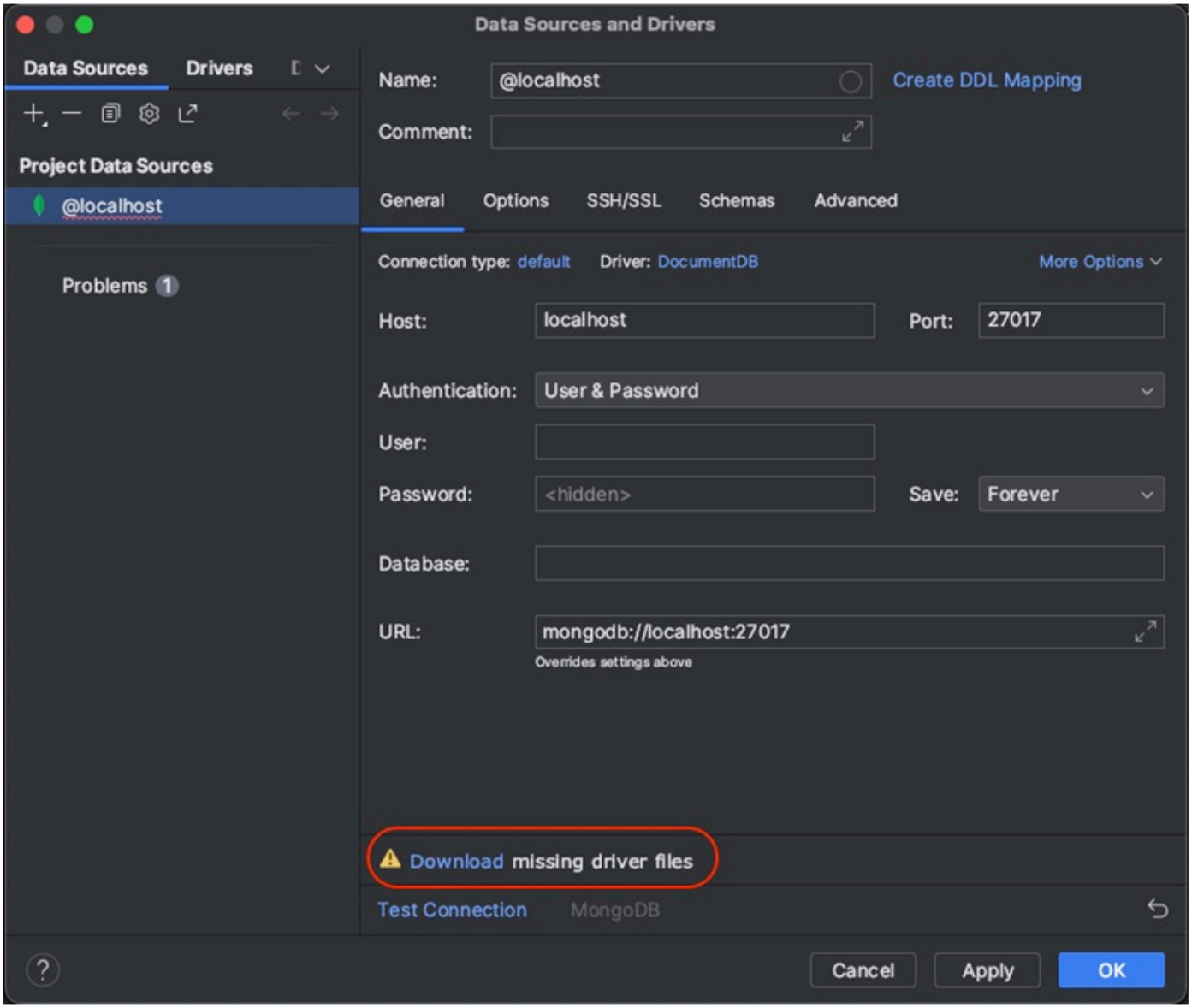
1. Lancez DataGrip sur votre ordinateur et créez un nouveau projet.



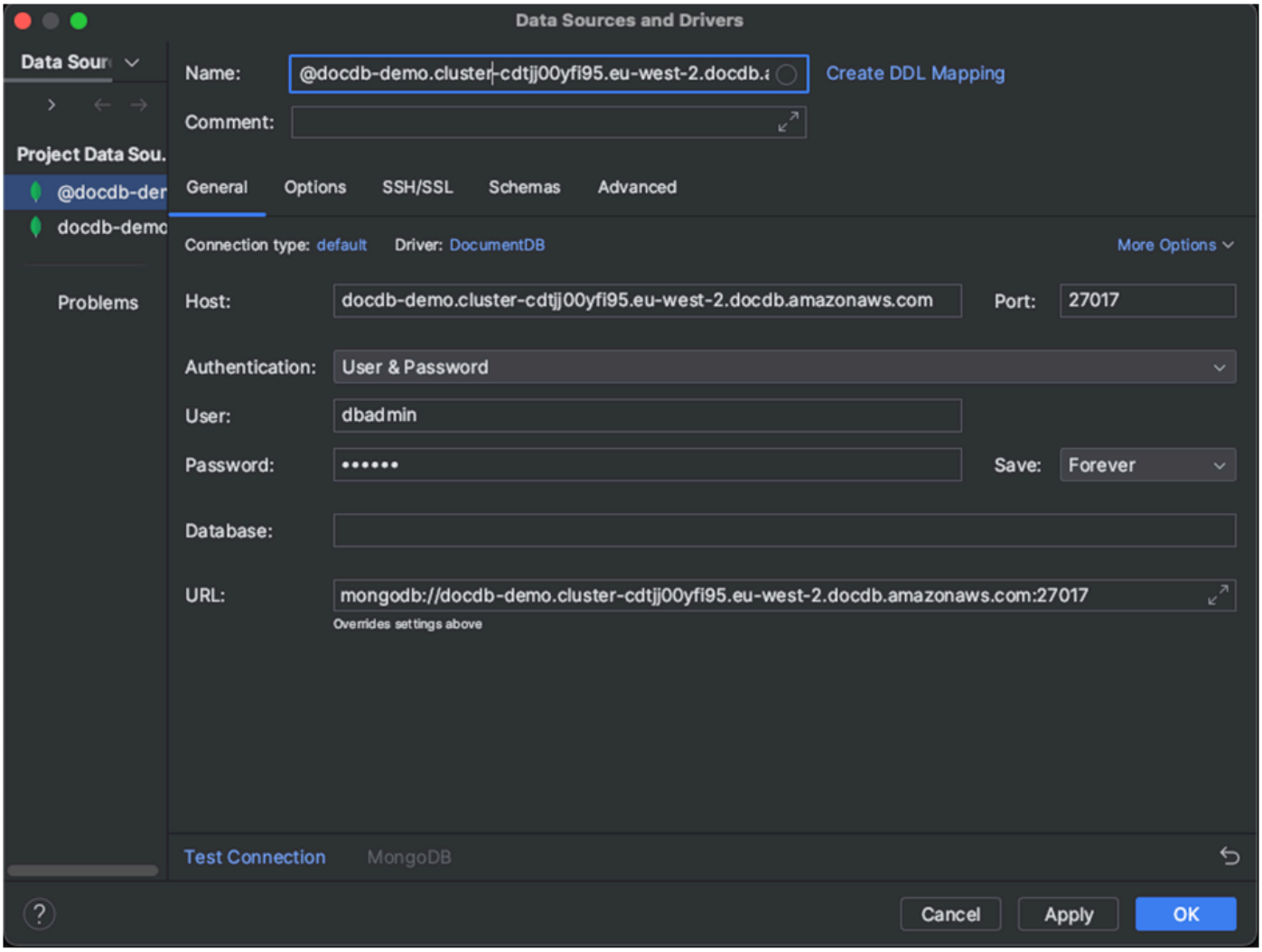
2. Ajoutez une nouvelle source de données en utilisant l'une des méthodes suivantes :
   a. Dans le menu principal, accédez à Fichier — Nouveau — Source de données et sélectionnez DocumentDB
   b. Dans l'explorateur de base de données, cliquez sur la nouvelle icône (+) dans la barre d'outils. Accédez à la source de données et sélectionnez DocumentDB.



3. Sur la page Sources de données de l'onglet Général, vérifiez s'il existe un lien Télécharger les fichiers de pilotes manquants en bas de la zone des paramètres de connexion. Cliquez sur ce lien pour télécharger les pilotes nécessaires pour interagir avec une base de données. Pour un lien de téléchargement direct, reportez-vous aux pilotes JetBrains JDBC.



4. Dans l'onglet Général, spécifiez les détails de la connexion :
   a. Dans le champ Host, spécifiez le point de terminaison du cluster Amazon DocumentDB.
   b. Le port est déjà défini sur 27017. Modifiez-le si votre cluster a été déployé sur un autre port.
   c. Pour Authentification, choisissez Utilisateur et mot de passe.
   d. Entrez votre nom d'utilisateur et votre mot de passe.
   e. Le champ Base de données est facultatif. Vous pouvez spécifier la base de données à laquelle vous souhaitez vous connecter.
   f. Le champ URL se complète automatiquement lorsque vous ajoutez les informations ci-dessus.



5. Dans l'onglet SSH/SSL, activez Utiliser le tunnel SSH, puis cliquez sur l'icône pour ouvrir la boîte de dialogue de configuration SSH. Entrez les informations suivantes :

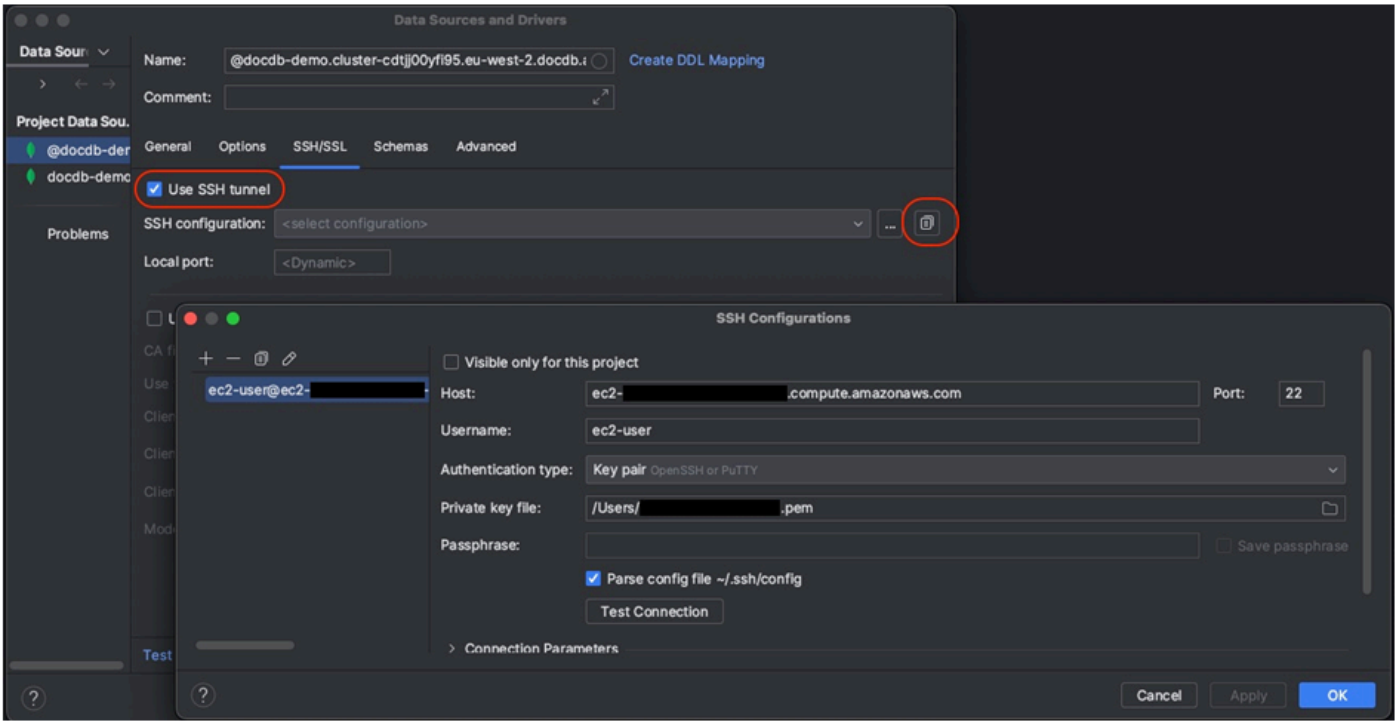
   a. dans le champ Host, entrez le nom d'hôte de votre instance Amazon EC2.

   b. Entrez le nom d'utilisateur et le mot de passe de votre instance Amazon EC2.

   c. Pour Type d'authentification, choisissez Key pair.

   d. Entrez votre fichier de clé privée.

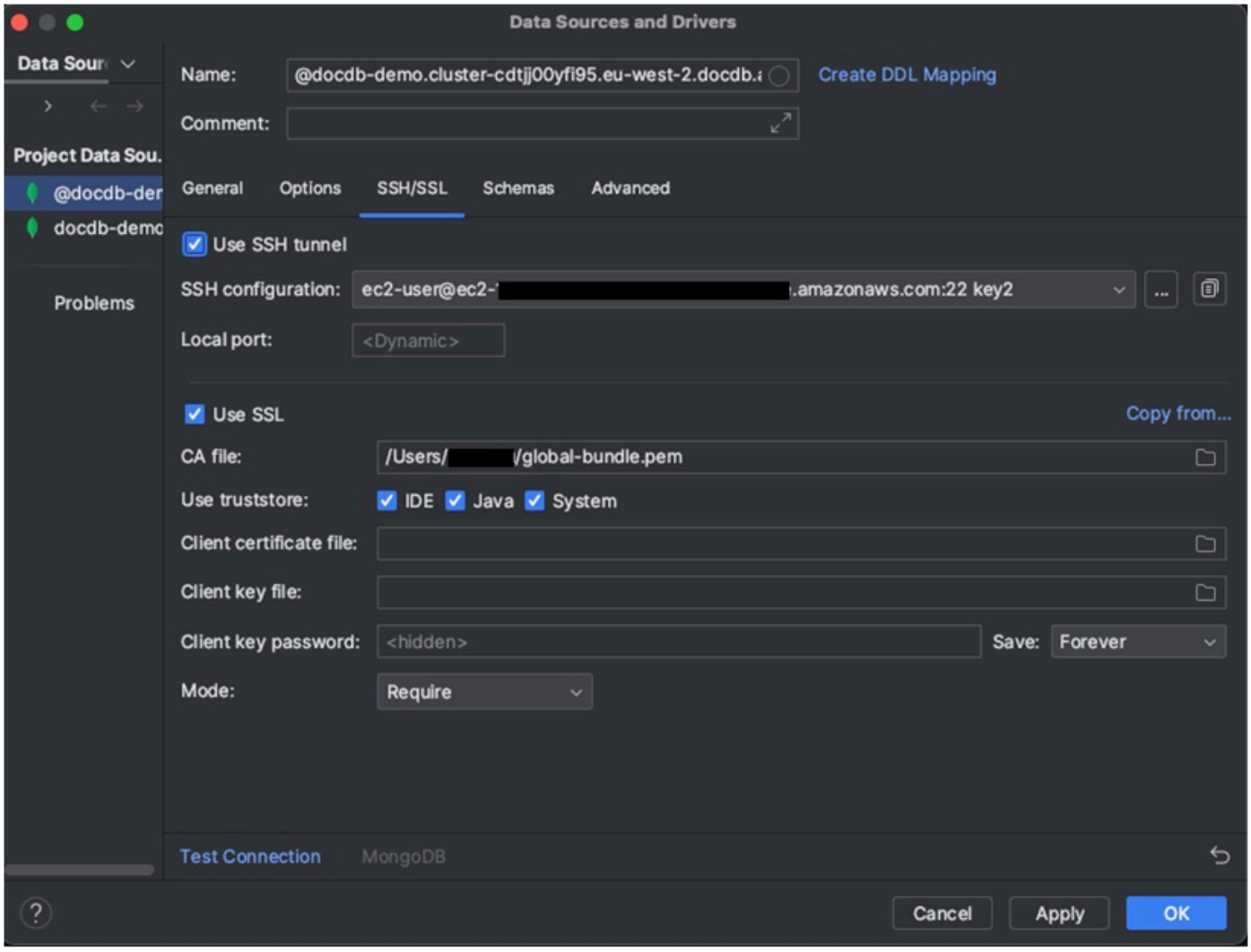
> **Note**
>
> Si vous utilisez l'option VPN, il n'est pas nécessaire de configurer le tunnel SSH.



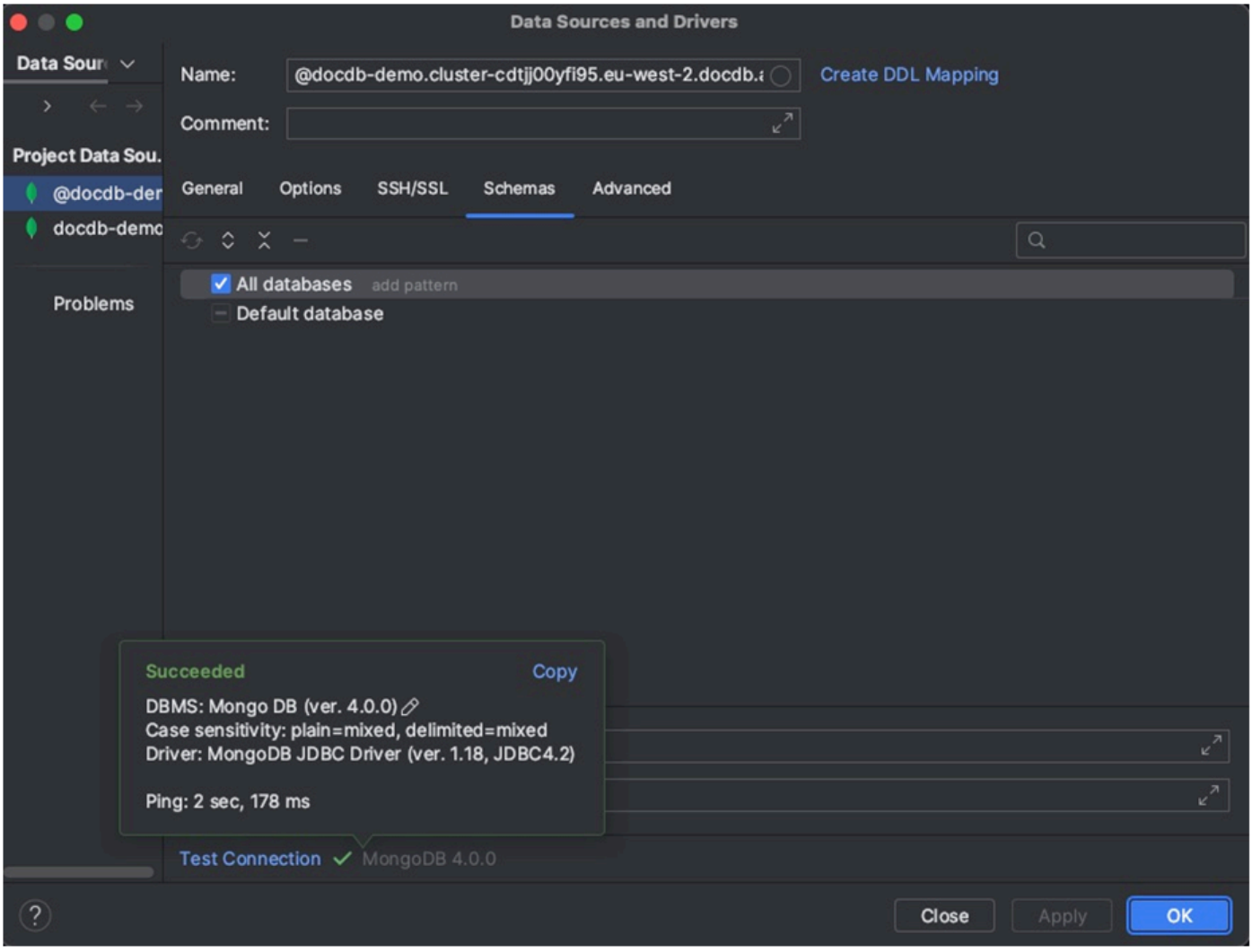
6. Dans l'onglet SSH/SSL, activez Utiliser SSL. Dans le champ Fichier CA, entrez l'emplacement du `global-bundle.pem` fichier sur votre ordinateur. Pour Mode, laissez l'option Exiger.

> **Note**
>
> Vous pouvez télécharger le certificat depuis cet emplacement ou avec cette commande : wget https://aws.amazon.com/https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem

> **Note**
>
> Si vous vous connectez au cluster élastique Amazon DocumentDB, il n'est pas nécessaire de spécifier le fichier CA. Laissez l'option Utiliser SSL cochée et toutes les autres options à leurs valeurs par défaut.



7. Dans l'onglet Schémas, choisissez Toutes les bases de données ou entrez le filtre « * : * » dans le champ Modèle de schéma. Cliquez sur le lien Tester la connexion pour tester la connexion.



8. Une fois la connexion testée avec succès, cliquez sur OK pour enregistrer la configuration de la source de données.

DataGrip fonctionnalités

DataGrip propose différentes fonctionnalités pour vous aider à travailler efficacement avec Amazon DocumentDB :

- Éditeur SQL : écrivez et exécutez des requêtes de type SQL sur vos collections DocumentDB à l'aide de l'éditeur SQL dans. DataGrip
- Générateur de requêtes visuel : utilisez le générateur de requêtes visuel pour créer des requêtes graphiquement sans écrire de code SQL.

- Gestion des schémas — Gérez facilement le schéma de votre base de données, notamment en créant, en modifiant et en supprimant des collections.
- Visualisation des données : visualisez et analysez vos données à l'aide des différents outils de visualisation disponibles dans DataGrip.
- Exportation et importation de données : transférez des données entre Amazon DocumentDB et d'autres bases de données à l'aide des fonctionnalités DataGrip d'exportation et d'importation.

Consultez la DataGrip documentation officielle pour obtenir des fonctionnalités plus avancées et des conseils sur l'utilisation d'Amazon DocumentDB et d'autres systèmes de base de données.

Connectez-vous à l'aide d'Amazon EC2

Cette section décrit comment configurer la connectivité entre un cluster Amazon DocumentDB et Amazon EC2 et comment accéder au cluster Amazon DocumentDB depuis l'instance Amazon EC2.

Il existe deux options pour configurer la connexion EC2 :

- Connectez automatiquement votre instance EC2 à une base de données Amazon DocumentDB : utilisez la fonctionnalité de connexion automatique de la console EC2 pour configurer automatiquement la connexion entre votre instance EC2 et une base de données Amazon DocumentDB nouvelle ou existante. Cette connexion permet au trafic de voyager entre l'instance EC2 et la base de données Amazon DocumentDB. Cette option est généralement utilisée pour tester et créer de nouveaux groupes de sécurité.
- Connectez manuellement votre instance EC2 à votre base de données Amazon DocumentDB : configurez la connexion entre votre instance EC2 et votre base de données Amazon DocumentDB en configurant et en attribuant manuellement les groupes de sécurité afin de reproduire la configuration créée par la fonctionnalité de connexion automatique. Cette option est généralement utilisée pour modifier des paramètres plus avancés et utiliser les groupes de sécurité existants.

Prérequis

Quelle que soit l'option, et avant de créer votre premier cluster Amazon DocumentDB, vous devez effectuer les opérations suivantes :

Créez un compte Amazon Web Services (AWS)

Avant de pouvoir commencer à utiliser Amazon DocumentDB, vous devez disposer d'un compte Amazon Web Services (AWS). Le AWS compte est gratuit. Vous payez uniquement les services et les ressources que vous utilisez.

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

Pour vous inscrire à un Compte AWS

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

   Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique 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. Pour des raisons de sécurité, attribuez un accès administratif à un utilisateur et utilisez uniquement l'utilisateur root pour effectuer les tâches nécessitant un accès utilisateur root.

Configurez éventuellement les autorisations AWS Identity and Access Management (IAM) nécessaires.

L'accès à la gestion des ressources Amazon DocumentDB telles que les clusters, les instances et les groupes de paramètres de cluster nécessite des informations d'identification AWS pouvant être utilisées pour authentifier vos demandes. Pour plus d'informations, consultez Identity and Access Management pour Amazon DocumentDB.

1. Dans la barre de recherche du AWS Management Console, tapez IAM et sélectionnez IAM dans le menu déroulant qui apparaît.
2. Une fois dans la console IAM, sélectionnez Utilisateurs dans le volet de navigation.
3. Sélectionnez votre nom d'utilisateur.
4. Cliquez sur le bouton Ajouter des autorisations.
5. Sélectionnez Attach existing policies directly (Attacher directement les politiques existantes).
6. Tapez `AmazonDocDBFullAccess` dans la barre de recherche et sélectionnez-la une fois qu'elle apparaît dans les résultats de recherche.

7. Cliquez sur le bouton bleu en bas qui indique Suivant : Réviser.
8. Cliquez sur le bouton bleu en bas indiquant Ajouter des autorisations.

Création d'un Amazon Virtual Private Cloud (Amazon VPC)

Selon l'endroit dans lequel Région AWS vous vous trouvez, il est possible qu'un VPC par défaut ait déjà été créé ou non. Si vous ne possédez pas de VPC par défaut, suivez l'étape 1 de la section Getting Started with Amazon VPC User Guide. Cela prendra moins de cinq minutes.

Connect Amazon EC2 automatiquement

Rubriques

- Connexion automatique d'une instance EC2 à une nouvelle base de données Amazon DocumentDB
- Connexion automatique d'une instance EC2 à une base de données Amazon DocumentDB existante
- Présentation de la connectivité automatique avec une instance EC2
- Affichage des ressources de calcul connectées

Avant de configurer une connexion entre une instance EC2 et une nouvelle base de données Amazon DocumentDB, assurez-vous de respecter les exigences décrites dans. Présentation de la connectivité automatique avec une instance EC2 Si vous modifiez les groupes de sécurité après avoir configuré la connectivité, ces modifications peuvent affecter la connexion entre l'instance EC2 et la base de données Amazon DocumentDB.

> **Note**
>
> Vous ne pouvez configurer automatiquement une connexion entre une instance EC2 et une base de données Amazon DocumentDB qu'à l'aide du. AWS Management Console Vous ne pouvez pas configurer de connexion automatiquement avec l'API AWS CLI ou Amazon DocumentDB.

Connexion automatique d'une instance EC2 à une nouvelle base de données Amazon DocumentDB

Le processus suivant suppose que vous avez effectué les étapes décrites dans la Prérequis rubrique.

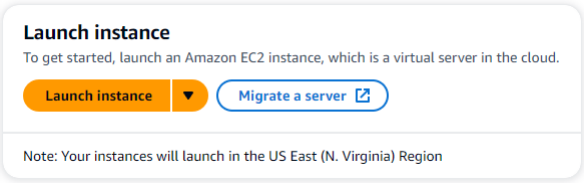
Étapes

- Étape 1 : créer une instance Amazon EC2
- Étape 2 : créer un cluster Amazon DocumentDB
- Étape 3 : Connectez-vous à votre instance Amazon EC2
- Étape 4 : Installation du shell Mongo
- Étape 5 : Gérer le protocole TLS d'Amazon DocumentDB
- Étape 6 : Connectez-vous à votre cluster Amazon DocumentDB
- Étape 7 : Insérer et interroger des données
- Étape 8 : Explorez

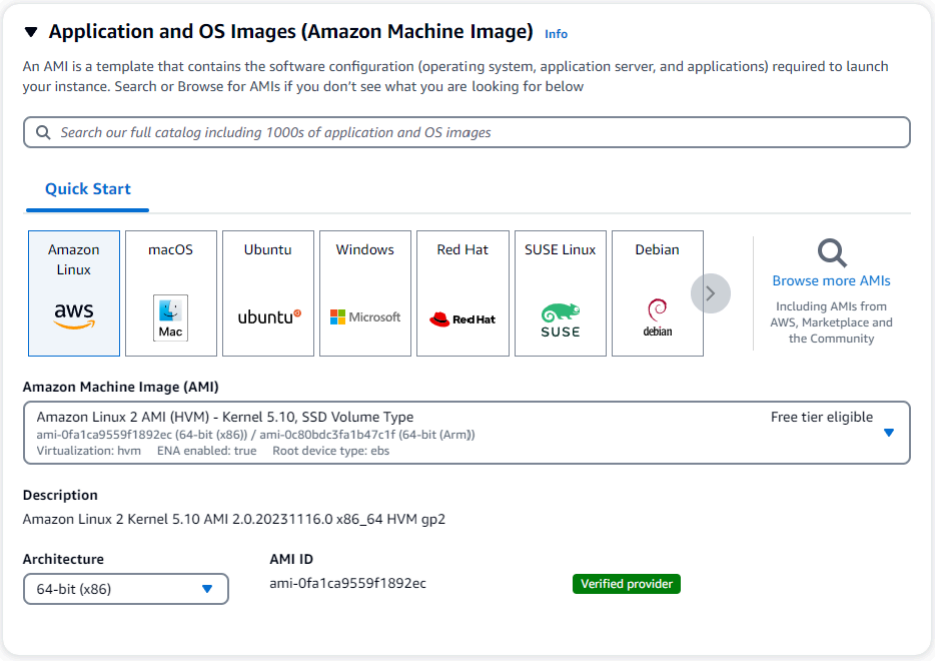
Étape 1 : créer une instance Amazon EC2

Au cours de cette étape, vous allez créer une instance Amazon EC2 dans la même région et dans le même Amazon VPC que vous utiliserez ultérieurement pour approvisionner votre cluster Amazon DocumentDB.

1. Sur la console Amazon EC2, choisissez Launch instance.

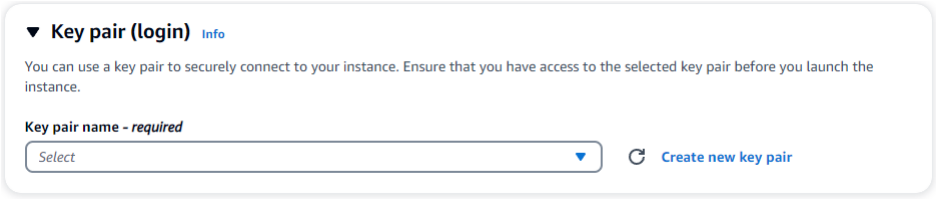


2. Entrez un nom ou un identifiant dans le champ Nom situé dans la section Nom et balises.
3. Dans la liste déroulante Amazon Machine Image (AMI), recherchez l'AMI Amazon Linux 2 et choisissez-la.



4. Recherchez et choisissez t3.micro dans la liste déroulante Type d'instance.

5. Dans la section Paire de clés (connexion), entrez l'identifiant d'une paire de clés existante ou choisissez Créer une nouvelle paire de clés.



Vous devez fournir une paire de clés Amazon EC2.

Si vous possédez une paire de clés Amazon EC2 :

a. Sélectionnez une paire de clés, choisissez votre paire de clés dans la liste.

b. Vous devez déjà disposer du fichier de clé privée (fichier .pem ou .ppk) pour vous connecter à votre instance Amazon EC2.

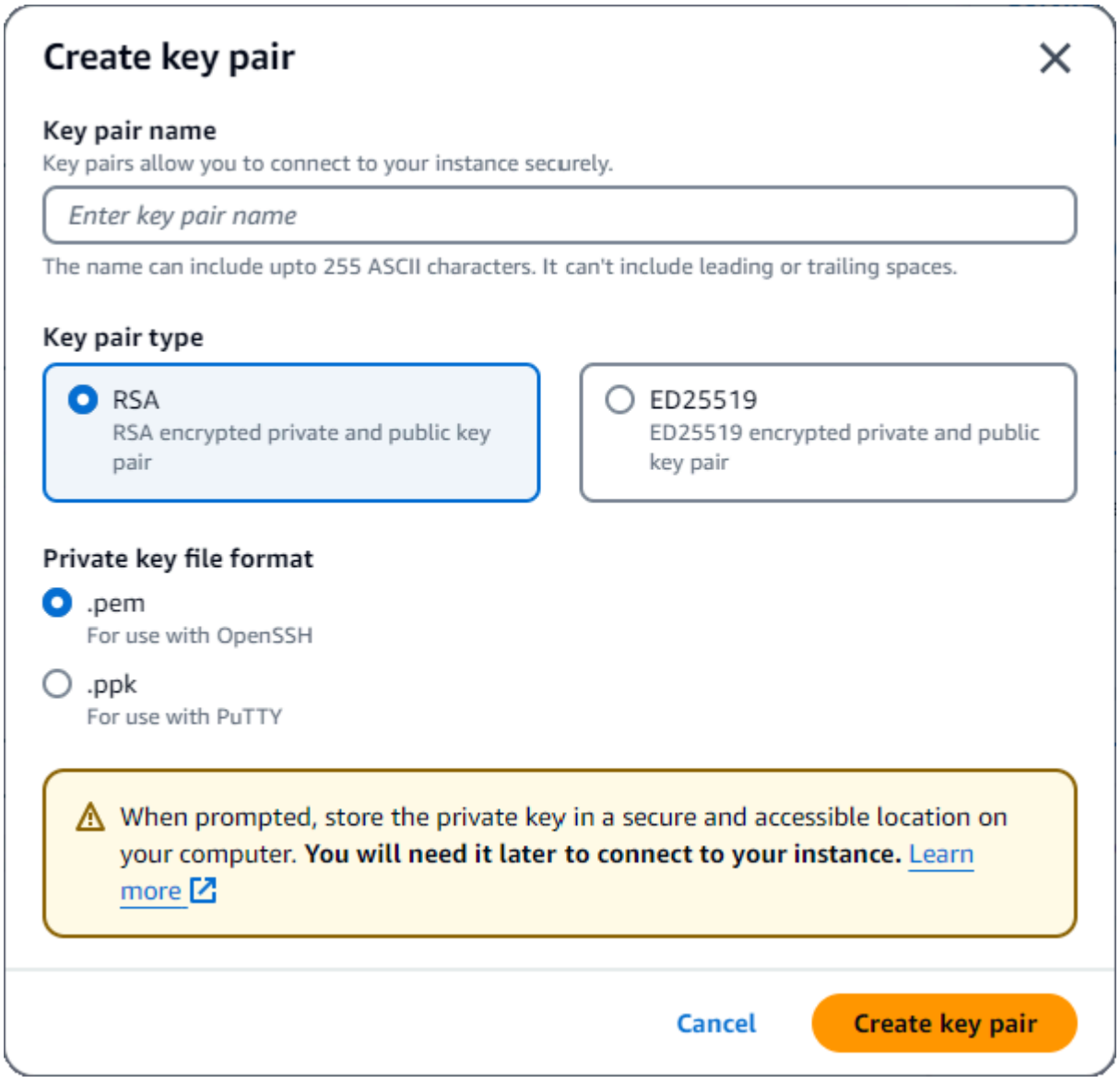
Si vous ne possédez pas de paire de clés Amazon EC2 :

a. Choisissez Créer une nouvelle paire de clés, la boîte de dialogue Créer une paire de clés apparaît.

b. Entrez un nom dans le champ Nom de la paire de clés.

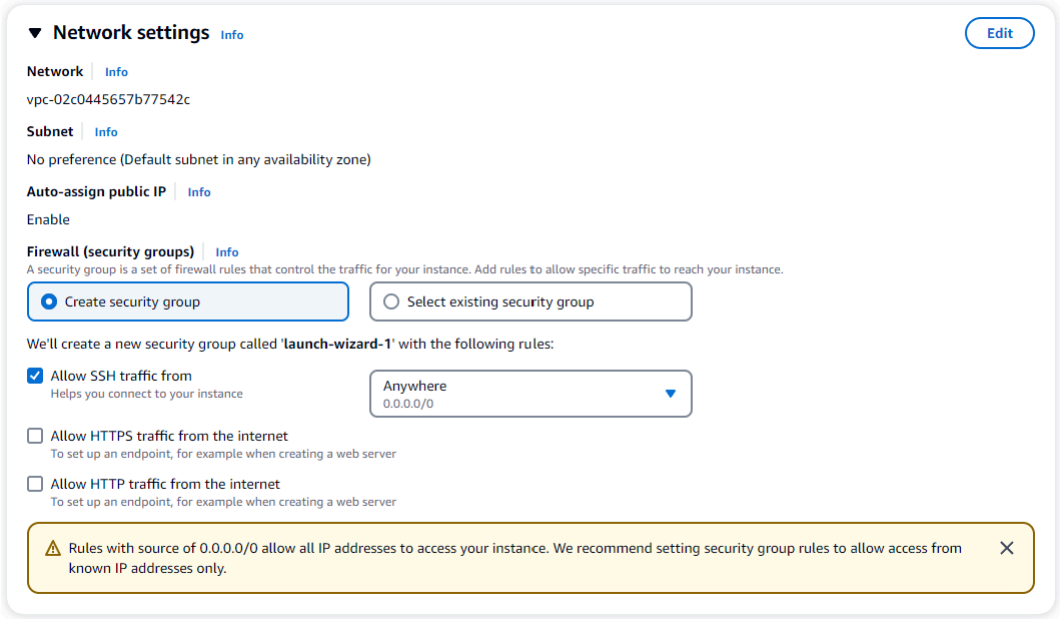
c. Choisissez le type de paire de clés et le format de fichier de clé privée.

d. Choisissez Créer une paire de clés.



> **Note**
>
> Pour des raisons de sécurité, nous vous recommandons vivement d'utiliser une paire de clés pour la connexion SSH et Internet à votre instance EC2.

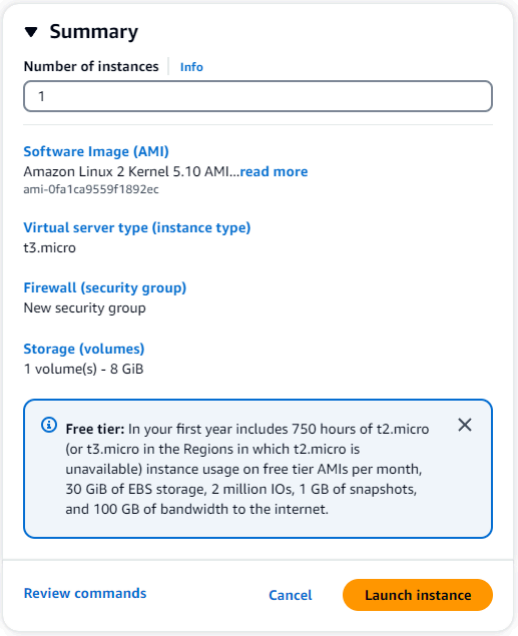
6. Facultatif : dans la section Paramètres réseau, sous Pare-feu (groupes de sécurité), choisissez Créer un groupe de sécurité ou Sélectionner un groupe de sécurité existant.



   Si vous avez choisi de sélectionner un groupe de sécurité existant, sélectionnez-en un dans la liste déroulante Groupes de sécurité communs.

   Si vous avez choisi de créer un nouveau groupe de sécurité, vérifiez toutes les règles d'autorisation du trafic qui s'appliquent à votre connectivité EC2.

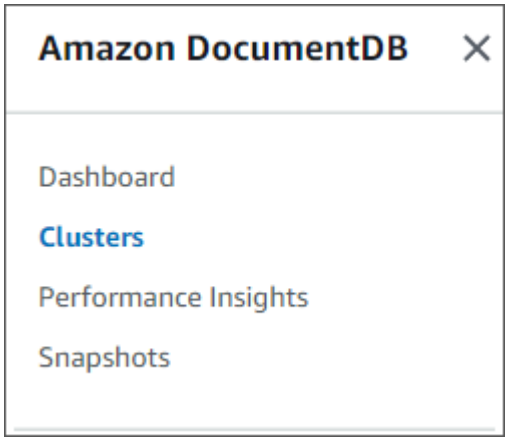
7. Dans la section Résumé, passez en revue votre configuration EC2 et choisissez Launch instance si c'est correct. Modifiez les groupes de sécurité.



Étape 2 : créer un cluster Amazon DocumentDB

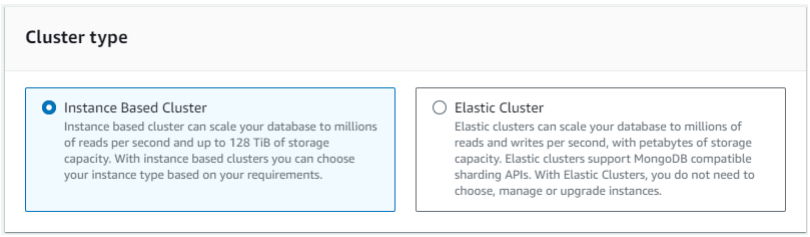
Pendant le provisionnement de l'instance Amazon EC2, vous allez créer votre cluster Amazon DocumentDB.

1. Accédez à la console Amazon DocumentDB et choisissez Clusters dans le volet de navigation.

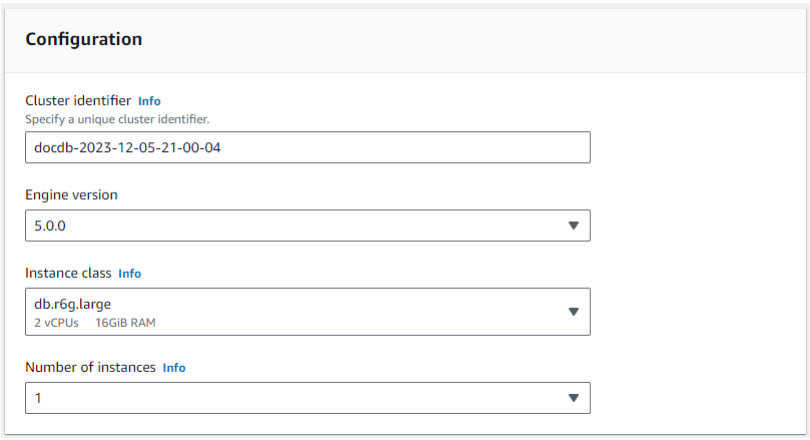


2. Choisissez Créer.

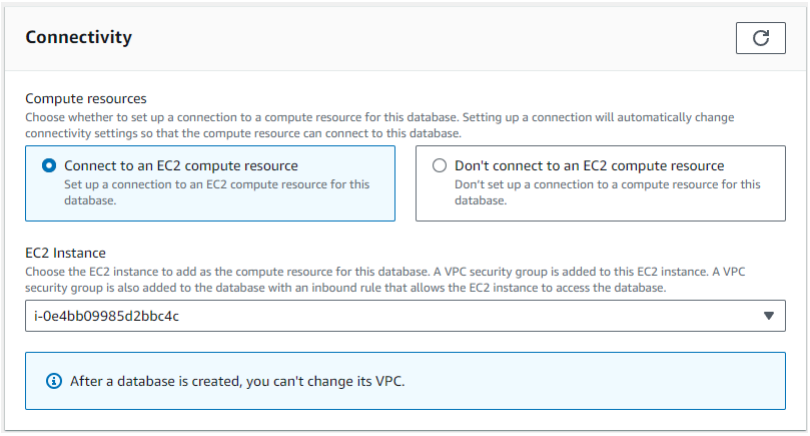
3. Conservez le paramètre Type de cluster à sa valeur par défaut, Cluster basé sur une instance.



4. Pour Nombre d'instances, choisissez 1. Cela minimisera les coûts. Conservez les autres paramètres par défaut.



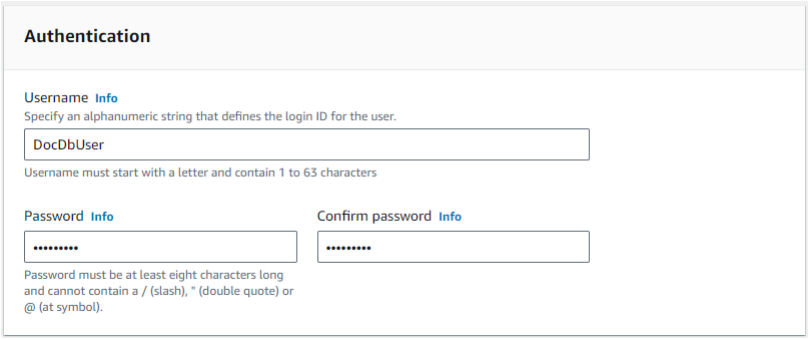
5. Pour Connectivity, choisissez Connect to an EC2 computing resource. Il s'agit de l'instance EC2 que vous avez créée à l'étape 1.



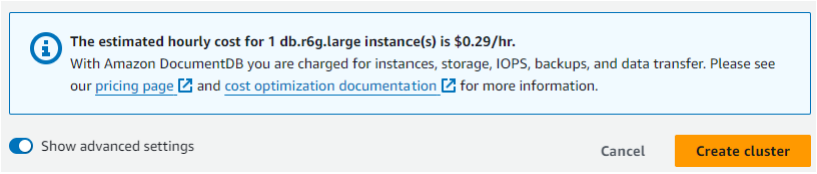
> **Note**
>
> La connexion à une ressource de calcul EC2 crée automatiquement un groupe de sécurité pour la connexion de votre ressource de calcul EC2 à votre cluster Amazon DocumentDB. Lorsque vous avez terminé de créer votre cluster et que vous souhaitez voir le groupe de sécurité nouvellement créé, accédez à la liste des clusters et choisissez l'identifiant de votre cluster. Dans l'onglet Connectivité

et sécurité, accédez à Groupes de sécurité et recherchez votre groupe sous Nom du groupe de sécurité (ID). Il contiendra le nom de votre cluster et ressemblera à ceci :`docdb-ec2-docdb-2023-12-11-21-33-41:i-0e4bb09985d2bbc4c (sg-0238e0b0bf0f73877)`.

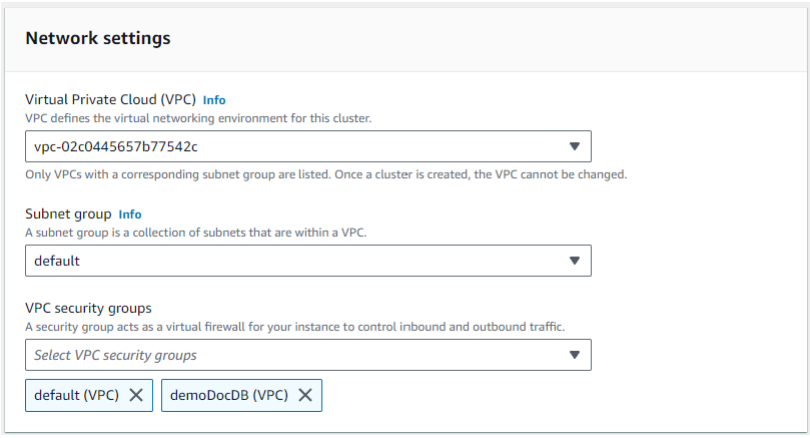
6. Pour l'authentification, entrez les informations de connexion. Important : Vous aurez besoin des informations de connexion pour authentifier votre cluster ultérieurement.



7. Activez Afficher les paramètres avancés.



8. Dans la section Paramètres réseau, pour les groupes de sécurité Amazon VPC, choisissez DemoDocDB.

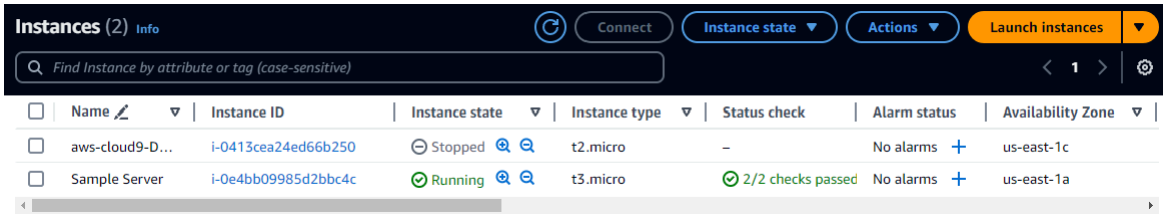


9. Choisissez Créer un cluster.

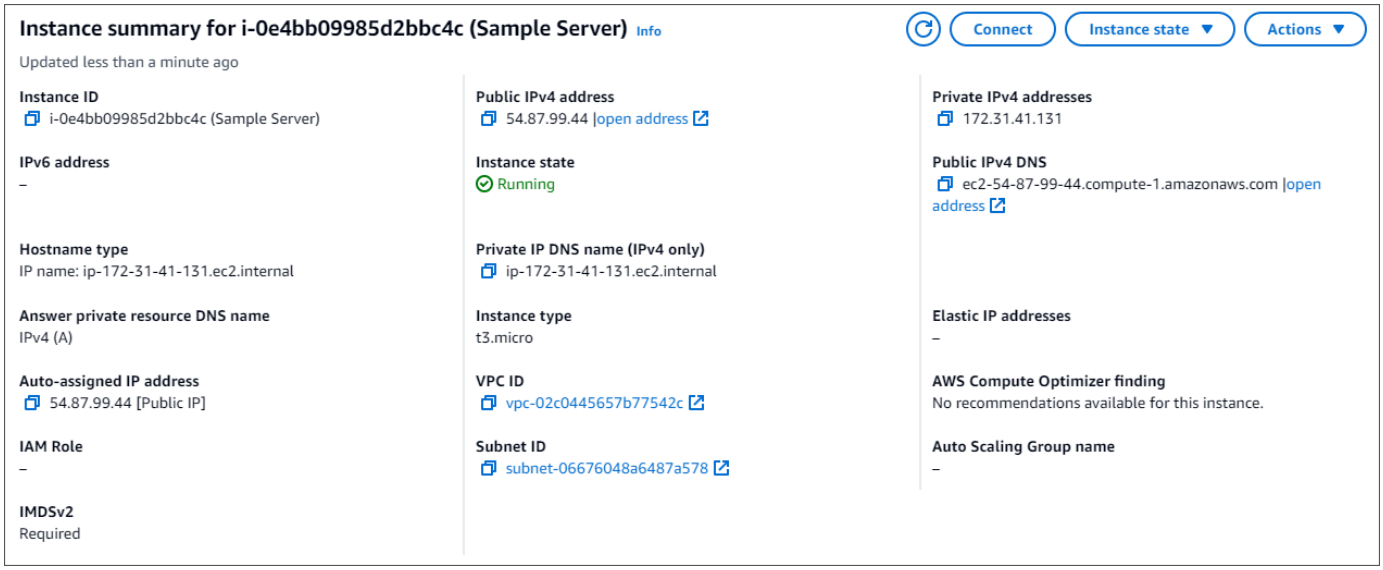
Étape 3 : Connectez-vous à votre instance Amazon EC2

Pour installer le shell mongo, vous devez d'abord vous connecter à votre instance Amazon EC2. L'installation du shell mongo vous permet de vous connecter à votre cluster Amazon DocumentDB et de l'interroger. Procédez comme suit :

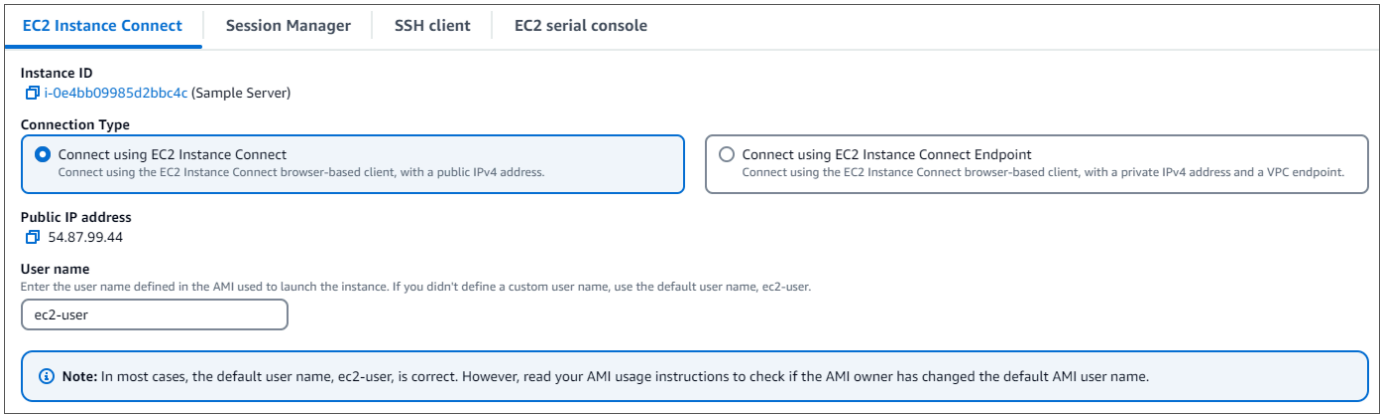
1. Sur la console Amazon EC2, accédez à vos instances et vérifiez si l'instance que vous venez de créer est en cours d'exécution. Si tel est le cas, sélectionnez l'instance en cliquant sur son ID.



2. Choisissez Se connecter.



3. Il existe quatre options à onglets pour votre méthode de connexion : Amazon EC2 Instance Connect, Session Manager, client SSH ou console série EC2. Vous devez en choisir un et suivre les instructions. Lorsque vous avez terminé, choisissez Connect.



> **Note**
>
> Si votre adresse IP a changé après le début de cette procédure, ou si vous revenez dans votre environnement ultérieurement, vous devez mettre à jour la règle de trafic entrant de votre groupe de `demoEC2` sécurité pour activer le trafic entrant depuis votre nouvelle adresse API.

Étape 4 : Installation du shell Mongo

Vous pouvez désormais installer le shell mongo, qui est un utilitaire de ligne de commande que vous utilisez pour connecter et interroger votre cluster Amazon DocumentDB. Suivez les instructions ci-dessous pour installer le shell mongo pour votre système d'exploitation.

On Amazon Linux

Pour installer le shell mongo sur Amazon Linux

1. Créez le fichier du référentiel. Sur la ligne de commande de votre instance EC2, exécutez la commande suivante :

```
echo -e "[mongodb-org-5.0] \nname=MongoDB Repository\nbaseurl=https://
repo.mongodb.org/yum/amazon/2/mongodb-org/5.0/x86_64/\ngpgcheck=1 \nenabled=1
 \ngpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc" | sudo tee /etc/
yum.repos.d/mongodb-org-5.0.repo
```

2. Une fois l'opération terminée, installez le shell mongo en exécutant la commande suivante :

```
sudo yum install -y mongodb-org-shell
```

On Ubuntu 18.04

Pour installer le shell mongo sur Ubuntu 18.04

1. Importez la clé publique qui sera utilisé par le système de gestion des packages.

   ```
   sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv
    2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5
   ```

2. Créez le fichier de liste `/etc/apt/sources.list.d/mongodb-org-3.6.list` pour MongoDB en utilisant la commande appropriée pour votre version d'Ubuntu.

   Ubuntu 18.04

   ```
   echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu xenial/
   mongodb-org/3.6 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-
   org-3.6.list
   ```

   > **Note**
   >
   > La commande ci-dessus installera le shell mongo 3.6 pour Bionic et Xenial.

3. Recharger la base de données de package locale à l'aide de la commande suivante :

   ```
   sudo apt-get update
   ```

4. Installez le shell MongoDB.

   ```
   sudo apt-get install -y mongodb-org-shell
   ```

Pour plus d'informations sur l'installation de versions antérieures de MongoDB sur votre système Ubuntu, consultez Installation de MongoDB Community Edition sur Ubuntu

On other operating systems

Pour installer le shell mongo sur d'autres systèmes d'exploitation, consultez les informations relatives à l'installation de MongoDB Community Edition, dans la documentation MongoDB.

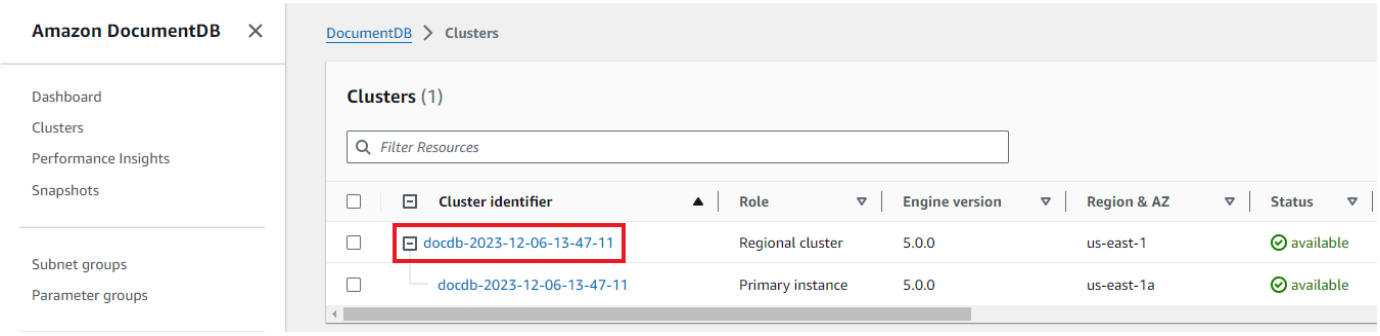
Étape 5 : Gérer le protocole TLS d'Amazon DocumentDB

Téléchargez le certificat CA pour Amazon DocumentDB avec le code suivant : `wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem`

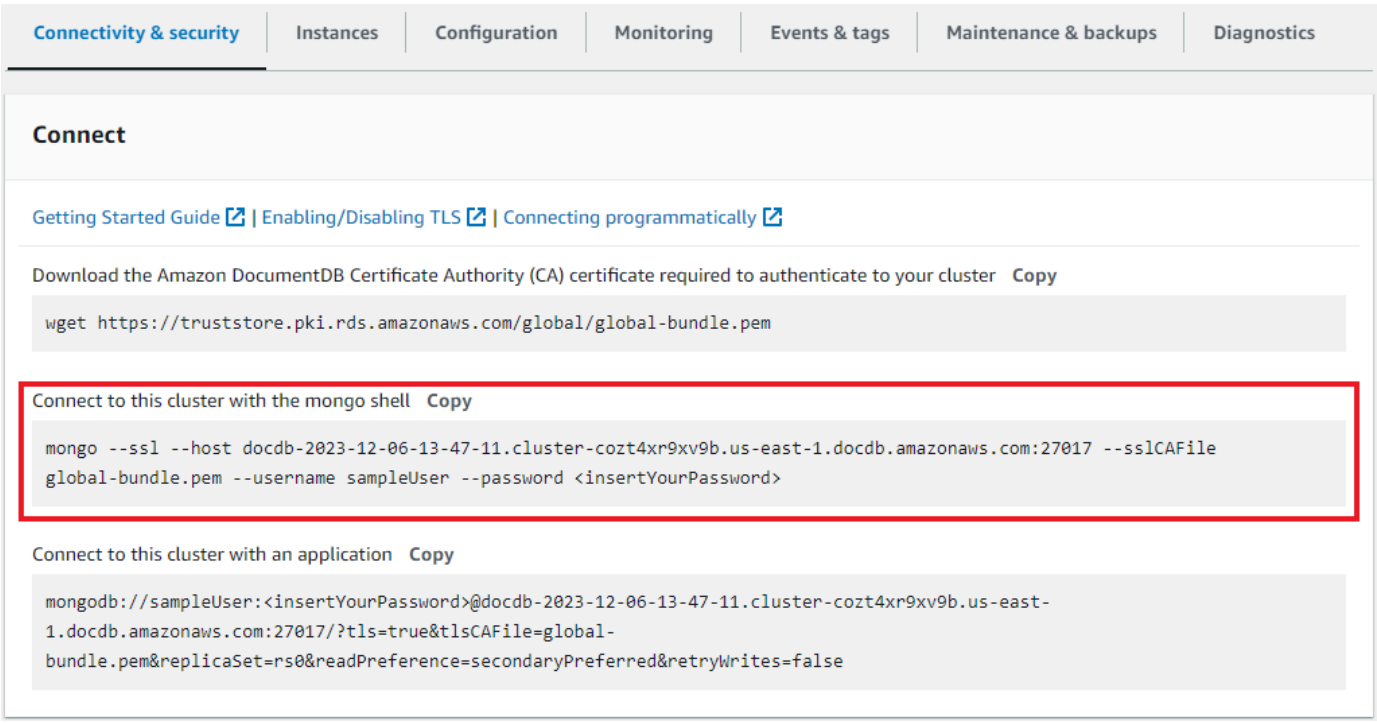
> **Note**
>
> Le protocole TLS (Transport Layer Security) est activé par défaut pour tous les nouveaux clusters Amazon DocumentDB. Pour plus d'informations, consultez Gestion des paramètres TLS du cluster Amazon DocumentDB.

Étape 6 : Connectez-vous à votre cluster Amazon DocumentDB

1. Sur la console Amazon DocumentDB, sous Clusters, localisez votre cluster. Choisissez le cluster que vous avez créé en cliquant sur l'identifiant du cluster.



2. Dans l'onglet Connectivité et sécurité, recherchez Connect to this cluster with the mongo shell dans le champ Connect :



Copiez la chaîne de connexion fournie et collez-la dans votre terminal.

Apportez-y les modifications suivantes :

a. Assurez-vous d'avoir le nom d'utilisateur correct dans la chaîne.

b. Omettez `<insertYourPassword>` afin que le shell mongo vous demande le mot de passe lorsque vous vous connectez.

Votre chaîne de connexion doit ressembler à ce qui suit :

```
mongo --ssl host docdb-2020-02-08-14-15-11.
cluster.region.docdb.amazonaws.com:27107 --sslCAFile global-bundle.pem
--username demoUser --password
```

3. Appuyez sur Entrée dans votre terminal. Vous êtes maintenant invité à saisir votre mot de passe. Entrez votre mot de passe.
4. Lorsque vous entrez votre mot de passe et que l'`rs0:PRIMARY>`invite s'affiche, vous êtes connecté avec succès à votre cluster Amazon DocumentDB.

Vous rencontrez des problèmes de connexion ? Consultez la section Résolution des problèmes liés à Amazon DocumentDB.

Étape 7 : Insérer et interroger des données

Maintenant que vous êtes connecté à votre cluster, vous pouvez exécuter quelques requêtes pour vous familiariser avec l'utilisation d'une base de données de documents.

1. Pour insérer un seul document, entrez les informations suivantes :

```
db.collection.insert({"hello":"DocumentDB"})
```

2. Vous obtenez le résultat suivant :

```
WriteResult({ "nInserted" : 1 })
```

3. Vous pouvez lire le document que vous avez écrit avec la `findOne()` commande (car il ne renvoie qu'un seul document). Entrez les informations suivantes :

```
db.collection.findOne()
```

4. Vous obtenez le résultat suivant :

```
{ "_id" : ObjectId("5e401fe56056fda7321fbd67"), "hello" :
"DocumentDB" }
```

5. Pour effectuer quelques requêtes supplémentaires, considérez un cas d'utilisation de profils de jeu. Tout d'abord, insérez quelques entrées dans une collection intituléeprofiles. Entrez les informations suivantes :

```
db.profiles.insertMany([
           { "_id" : 1, "name" : "Matt", "status": "active", "level": 12,
 "score":202},
           { "_id" : 2, "name" : "Frank", "status": "inactive", "level": 2,
 "score":9},
           { "_id" : 3, "name" : "Karen", "status": "active", "level": 7,
 "score":87},
           { "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
 "score":27}
           ])
```

6. Vous obtenez le résultat suivant :

```
{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4 ] }
```

7. Utilisez la `find()` commande pour renvoyer tous les documents de la collection de profils. Entrez les informations suivantes :

```
db.profiles.find()
```

8. Vous obtiendrez une sortie qui correspondra aux données que vous avez saisies à l'étape 5.
9. Utilisez une requête pour un seul document à l'aide d'un filtre. Entrez les informations suivantes :

```
db.profiles.find({name: "Katie"})
```

10. Vous devriez récupérer cette sortie :

```
{ "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
"score":27}
```

11. Essayons maintenant de trouver un profil et de le modifier à l'aide de la `findAndModify` commande. Nous allons donner dix points supplémentaires à l'utilisateur Matt avec le code suivant :

```
db.profiles.findAndModify({
    query: { name: "Matt", status: "active"},
    update: { $inc: { score: 10 } }
})
```

12. Vous obtenez le résultat suivant (notez que son score n'a pas encore augmenté) :

```
{
"_id" : 1,
"name" : "Matt",
"status" : "active",
"level" : 12,
"score" : 202
}
```

13. Vous pouvez vérifier que son score a changé avec la requête suivante :

```
db.profiles.find({name: "Matt"})
```

14. Vous obtenez le résultat suivant :

```
{ "_id" : 1, "name" : "Matt", "status" : "active", "level" : 12,
"score" : 212 }
```

Étape 8 : Explorez

Félicitations ! Vous avez terminé avec succès le guide de démarrage rapide d'Amazon DocumentDB.

Quelle est la prochaine étape ? Découvrez comment tirer pleinement parti de cette puissante base de données avec certaines de ses fonctionnalités populaires :

- Gestion d'Amazon DocumentDB
- Dimensionnement
- Sauvegarde et restauration

> **Note**
>
> Pour réduire les coûts, vous pouvez soit arrêter votre cluster Amazon DocumentDB afin de réduire les coûts, soit supprimer le cluster. Par défaut, après 30 minutes d'inactivité, votre AWS Cloud9 environnement arrête l'instance Amazon EC2 sous-jacente.

Connexion automatique d'une instance EC2 à une base de données Amazon DocumentDB existante

La procédure suivante suppose que vous disposez d'un cluster Amazon DocumentDB et d'une instance Amazon EC2 existants.

Accédez à votre cluster Amazon DocumentDB et configurez la connexion Amazon EC2

1. Accédez à votre cluster Amazon DocumentDB.

    a. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.

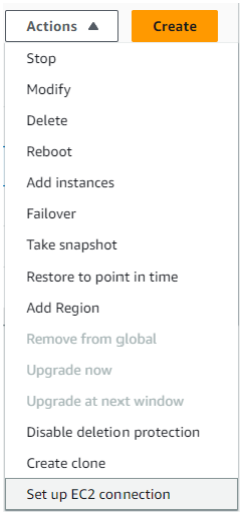
    b. Dans le panneau de navigation, choisissez Clusters.

> **Tip**
>
> Si vous ne voyez pas le volet de navigation sur le côté gauche de votre écran, choisissez l'icône de menu (≡) dans le coin supérieur gauche de la page.

c. Spécifiez le cluster que vous souhaitez en cliquant sur le bouton situé à gauche du nom du cluster.

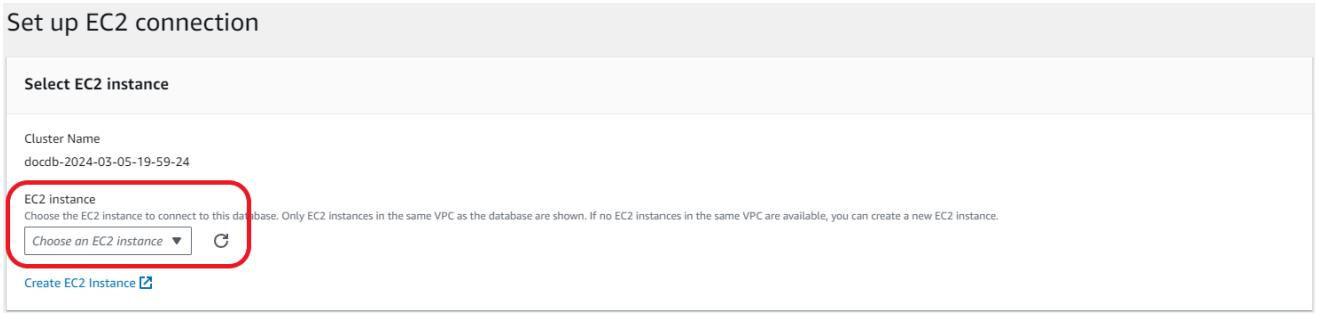
2. Configurez la connexion Amazon EC2.

a. Choisissez Actions, puis sélectionnez Configurer la connexion EC2.



La boîte de dialogue Configurer la connexion EC2 s'affiche.

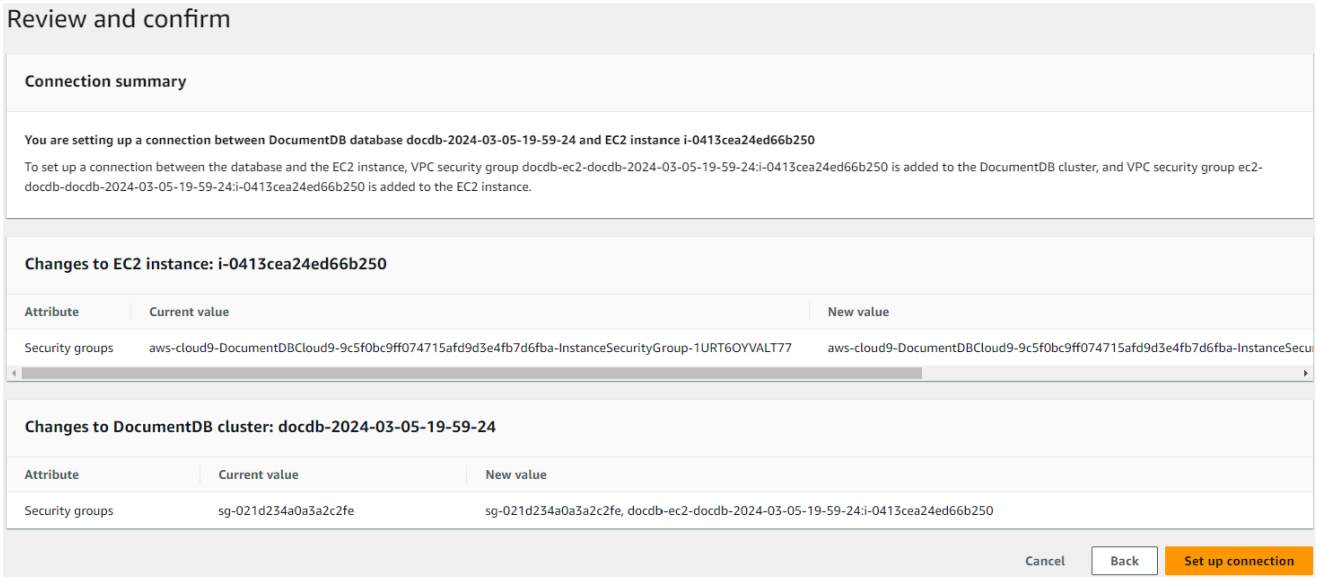
b. Dans le champ instance EC2, choisissez l'instance EC2 que vous souhaitez connecter à votre cluster.



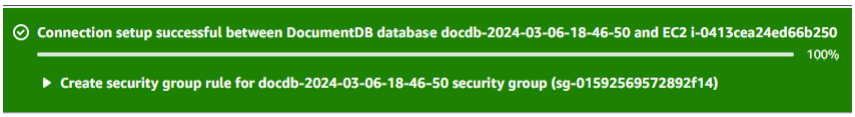
c. Choisissez Continuer.

La boîte de dialogue Vérifier et confirmer apparaît.

d. Assurez-vous que les modifications sont correctes. Choisissez ensuite Configurer la connexion.



En cas de succès, la vérification suivante apparaît :



Présentation de la connectivité automatique avec une instance EC2

Lorsque vous configurez une connexion entre une instance EC2 et une base de données Amazon DocumentDB, Amazon DocumentDB configure automatiquement le groupe de sécurité VPC pour votre instance EC2 et pour votre base de données Amazon DocumentDB.

Les conditions requises pour connecter une instance EC2 à une base de données Amazon DocumentDB sont les suivantes :

- L'instance EC2 doit exister dans le même VPC que la base de données Amazon DocumentDB.

  S'il n'y a pas d'instances EC2 dans le même VPC, la console fournit un lien pour en créer une.
- L'utilisateur qui configure la connectivité doit avoir les autorisations nécessaires pour effectuer les opérations Amazon EC2 suivantes :
  - `ec2:AuthorizeSecurityGroupEgress`

- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:CreateSecurityGroup`
- `ec2:DescribeInstances`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeSecurityGroups`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:RevokeSecurityGroupEgress`

Si l'instance de base de données et l'instance EC2 se trouvent dans des zones de disponibilité différentes, votre compte peut être confronté à des coûts croisés entre zones de disponibilité.

Lorsque vous configurez une connexion à une instance EC2, Amazon DocumentDB agit conformément à la configuration actuelle des groupes de sécurité associés à la base de données Amazon DocumentDB et à l'instance EC2, comme décrit dans le tableau suivant :

Configuration actuelle du groupe de sécurité Amazon DocumentDB	Configuration du groupe de sécurité EC2 actuel	Action Amazon DocumentDB
Un ou plusieurs groupes de sécurité sont associés à la base de données Amazon DocumentDB dont le nom correspond au modèle. `DocumentDB-ec2-n` Un groupe de sécurité qui correspond au modèle n'a pas été modifié. Ce groupe de sécurité comprend une seule règle entrante avec le groupe de sécurité du VPC de l'instance EC2 comme source.	Un ou plusieurs groupes de sécurité sont associés à l'instance EC2 dont le nom correspond au modèle `DocumentDB-ec2-n` (où n est un nombre). Un groupe de sécurité qui correspond au modèle n'a pas été modifié. Ce groupe de sécurité ne possède qu'une seule règle sortante avec le groupe de sécurité VPC de la base de données Amazon DocumentDB comme source.	Amazon DocumentDB n'entreprend aucune action. Une connexion a déjà été configurée automatiquement entre l'instance EC2 et la base de données Amazon DocumentDB. Comme une connexion existe déjà entre l'instance EC2 et la base de données Amazon DocumentDB, les groupes de sécurité ne sont pas modifiés.

Configuration actuelle du groupe de sécurité Amazon DocumentDB	Configuration du groupe de sécurité EC2 actuel	Action Amazon DocumentDB
L'une des conditions suivantes s'applique : • Aucun groupe de sécurité n'est associé à la base de données Amazon DocumentDB dont le nom correspond au modèle. `DocumentDB-ec2-n` • Un ou plusieurs groupes de sécurité sont associés à Amazon DocumentDB dont le nom correspond au modèle. `DocumentDB-ec2-n` Cependant, Amazon DocumentDB ne peut utiliser aucun de ces groupes de sécurité pour la connexion avec l'instance EC2. Amazon DocumentDB ne peut pas utiliser un groupe de sécurité qui n'a pas de règle entrante avec le groupe de sécurité VPC de l'instance EC2 comme source. Amazon DocumentDB ne peut pas non plus utiliser un groupe de sécurité qui a été modifié. Des exemples de modifications incluent l'ajout d'une règle ou la	L'une des conditions suivantes s'applique : • Aucun groupe de sécurité n'est associé à l'instanc e EC2 avec un nom qui correspond au modèle `ec2-DocumentDB-n` . • Un ou plusieurs groupes de sécurité sont associés à l'instance EC2 avec un nom qui correspond au modèle `ec2-Docum entDB-n` . Cependant , Amazon DocumentDB ne peut utiliser aucun de ces groupes de sécurité pour la connexion à la base de données Amazon DocumentDB. Amazon DocumentDB ne peut pas utiliser un groupe de sécurité qui n'a pas de règle sortante avec le groupe de sécurité VPC de la base de données Amazon DocumentDB comme source. Amazon DocumentDB ne peut pas non plus utiliser un groupe de sécurité qui a été modifié.	Action Amazon DocumentDB : créer de nouveaux groupes de sécurité

Configuration actuelle du groupe de sécurité Amazon DocumentDB	Configuration du groupe de sécurité EC2 actuel	Action Amazon DocumentDB
modification du port d'une règle existante.		
Un ou plusieurs groupes de sécurité sont associés à la base de données Amazon DocumentDB dont le nom correspond au modèle. `DocumentDB-ec2-n` Un groupe de sécurité qui correspond au modèle n'a pas été modifié. Ce groupe de sécurité comprend une seule règle entrante avec le groupe de sécurité du VPC de l'instance EC2 comme source.	Un ou plusieurs groupes de sécurité sont associés à l'instance EC2 avec un nom qui correspond au modèle `ec2-DocumentDB-n`. Cependant, Amazon DocumentDB ne peut utiliser aucun de ces groupes de sécurité pour la connexion à la base de données Amazon DocumentDB. Amazon DocumentDB ne peut pas utiliser un groupe de sécurité qui n'a pas de règle sortante avec le groupe de sécurité VPC de la base de données Amazon DocumentDB comme source. Amazon DocumentDB ne peut pas non plus utiliser un groupe de sécurité qui a été modifié.	Action Amazon DocumentDB : créer de nouveaux groupes de sécurité

Configuration actuelle du groupe de sécurité Amazon DocumentDB	Configuration du groupe de sécurité EC2 actuel	Action Amazon DocumentDB
Un ou plusieurs groupes de sécurité sont associés à la base de données Amazon DocumentDB dont le nom correspond au modèle. `DocumentDB-ec2-n` Un groupe de sécurité qui correspond au modèle n'a pas été modifié. Ce groupe de sécurité comprend une seule règle entrante avec le groupe de sécurité du VPC de l'instance EC2 comme source.	Il existe un groupe de sécurité EC2 valide pour la connexion , mais il n'est pas associé à l'instance EC2. Le nom de ce groupe de sécurité correspon d au modèle `DocumentD B-ec2-n` . Il n'a pas été modifié. Il n'a qu'une seule règle sortante avec le groupe de sécurité VPC de la base de données Amazon DocumentD B comme source.	Action Amazon DocumentDB : associer un groupe de sécurité EC2

Configuration actuelle du groupe de sécurité Amazon DocumentDB	Configuration du groupe de sécurité EC2 actuel	Action Amazon DocumentDB
L'une des conditions suivantes s'applique : • Aucun groupe de sécurité n'est associé à la base de données Amazon DocumentDB dont le nom correspond au modèle. `DocumentDB-ec2-n` • Un ou plusieurs groupes de sécurité sont associés à la base de données Amazon DocumentDB dont le nom correspond au modèle. `DocumentDB-ec2-n` Cependant, Amazon DocumentDB ne peut utiliser aucun de ces groupes de sécurité pour la connexion avec l'instance EC2. Amazon DocumentDB ne peut pas utiliser un groupe de sécurité qui n'a pas de règle entrante avec le groupe de sécurité VPC de l'instance EC2 comme source. Amazon DocumentDB ne peut pas non plus utiliser le groupe de sécurité qui a été modifié.	Un ou plusieurs groupes de sécurité sont associés à l'instance EC2 avec un nom qui correspond au modèle `DocumentDB-ec2-n` . Un groupe de sécurité qui correspond au modèle n'a pas été modifié. Ce groupe de sécurité ne possède qu'une seule règle sortante avec le groupe de sécurité VPC de la base de données Amazon DocumentDB comme source.	Action Amazon DocumentDB : créer de nouveaux groupes de sécurité

Action Amazon DocumentDB : créer de nouveaux groupes de sécurité

Amazon DocumentDB effectue les actions suivantes :

- Crée un nouveau groupe de sécurité qui correspond au modèle `DocumentDB-ec2-n`. Ce groupe de sécurité comprend une règle entrante avec le groupe de sécurité du VPC de l'instance EC2 comme source. Ce groupe de sécurité est associé à la base de données Amazon DocumentDB et permet à l'instance EC2 d'accéder à la base de données Amazon DocumentDB.
- Crée un nouveau groupe de sécurité qui correspond au modèle `ec2-DocumentDB-n`. Ce groupe de sécurité possède une règle sortante dont le groupe de sécurité VPC de la base de données Amazon DocumentDB est la source. Ce groupe de sécurité est associé à l'instance EC2 et permet à celle-ci d'envoyer du trafic vers la base de données Amazon DocumentDB.

Action Amazon DocumentDB : associer un groupe de sécurité EC2

Amazon DocumentDB associe le groupe de sécurité EC2 existant et valide à l'instance EC2. Ce groupe de sécurité permet à l'instance EC2 d'envoyer du trafic vers la base de données Amazon DocumentDB.

Affichage des ressources de calcul connectées

Vous pouvez utiliser le AWS Management Console pour afficher les ressources de calcul connectées à une base de données Amazon DocumentDB. Les ressources affichées comprennent les connexions de ressources de calcul qui ont été configurées automatiquement. Vous pouvez configurer automatiquement la connectivité avec les ressources de calcul de la manière suivante :

- Vous pouvez sélectionner la ressource de calcul lorsque vous créez la base de données. Pour plus d'informations, voir Création Création d'un cluster Amazon DocumentDB d'un cluster de base de données multi-AZ.
- Vous pouvez configurer la connectivité entre une base de données existante et une ressource de calcul. Pour plus d’informations, consultez Connect Amazon EC2 automatiquement.

Les ressources de calcul répertoriées n'incluent pas celles qui ont été connectées manuellement à la base de données. Par exemple, vous pouvez autoriser une ressource de calcul à accéder manuellement à une base de données en ajoutant une règle au groupe de sécurité du VPC associé à la base de données.

Pour qu'une ressource de calcul soit répertoriée, les conditions suivantes doivent s'appliquer :

- Le nom du groupe de sécurité associé à la ressource de calcul correspond au modèle `ec2-DocumentDB-n` (où n est un nombre).
- Le groupe de sécurité associé à la ressource de calcul possède une règle sortante dont la plage de ports est définie sur le port utilisé par la base de données Amazon DocumentDB.
- Le groupe de sécurité associé à la ressource de calcul possède une règle sortante dont la source est définie sur un groupe de sécurité associé à la base de données Amazon DocumentDB.
- Le nom du groupe de sécurité associé à la base de données Amazon DocumentDB correspond au modèle `DocumentDB-ec2-n` (où n est un nombre).
- Le groupe de sécurité associé à la base de données Amazon DocumentDB possède une règle entrante dont la plage de ports est définie sur le port utilisé par la base de données Amazon DocumentDB.
- Le groupe de sécurité associé à la base de données Amazon DocumentDB possède une règle entrante dont la source est définie sur un groupe de sécurité associé à la ressource de calcul.

Pour afficher les ressources de calcul connectées à une base de données Amazon DocumentDB

1. Connectez-vous à la AWS Management Console console Amazon DocumentDB et ouvrez-la à l'adresse https://console.aws.amazon.com/docdb.
2. Dans le volet de navigation, choisissez Databases, puis le nom de la base de données Amazon DocumentDB.
3. Dans l'onglet Connectivité et sécurité, consultez les ressources de calcul dans la section Ressources de calcul connectées.

Connect Amazon EC2 manuellement

Rubriques

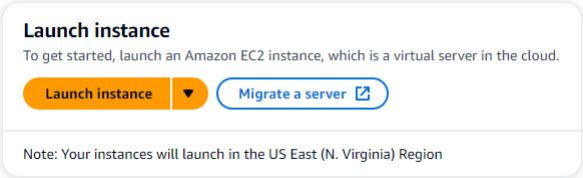
- Étape 8 : Insérer et interroger des données
- Étape 9 : Explorez

Les étapes suivantes supposent que vous avez terminé les étapes décrites dans la Prérequis rubrique.

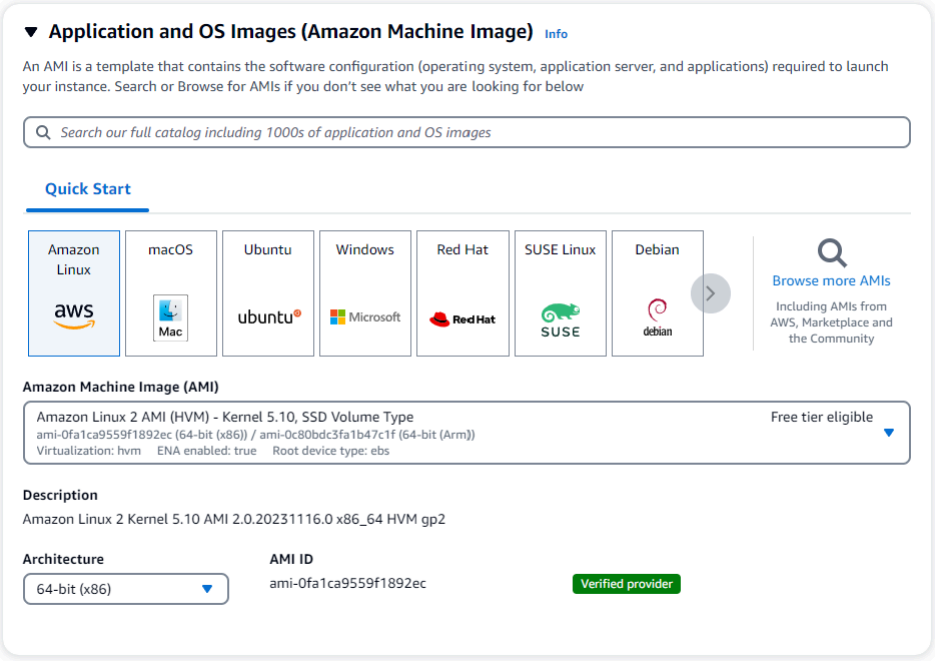
Étape 1 : créer une instance Amazon EC2

Au cours de cette étape, vous allez créer une instance Amazon EC2 dans la même région et dans le même Amazon VPC que vous utiliserez ultérieurement pour approvisionner votre cluster Amazon DocumentDB.

1. Sur la console Amazon EC2, choisissez Launch instance.

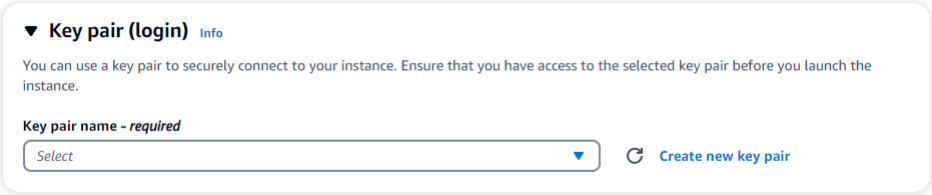


2. Entrez un nom ou un identifiant dans le champ Nom situé dans la section Nom et balises.
3. Dans la liste déroulante Amazon Machine Image (AMI), recherchez l'AMI Amazon Linux 2 et choisissez-la.



4. Recherchez et choisissez t3.micro dans la liste déroulante Type d'instance.

5. Dans la section Paire de clés (connexion), entrez l'identifiant d'une paire de clés existante ou choisissez Créer une nouvelle paire de clés.



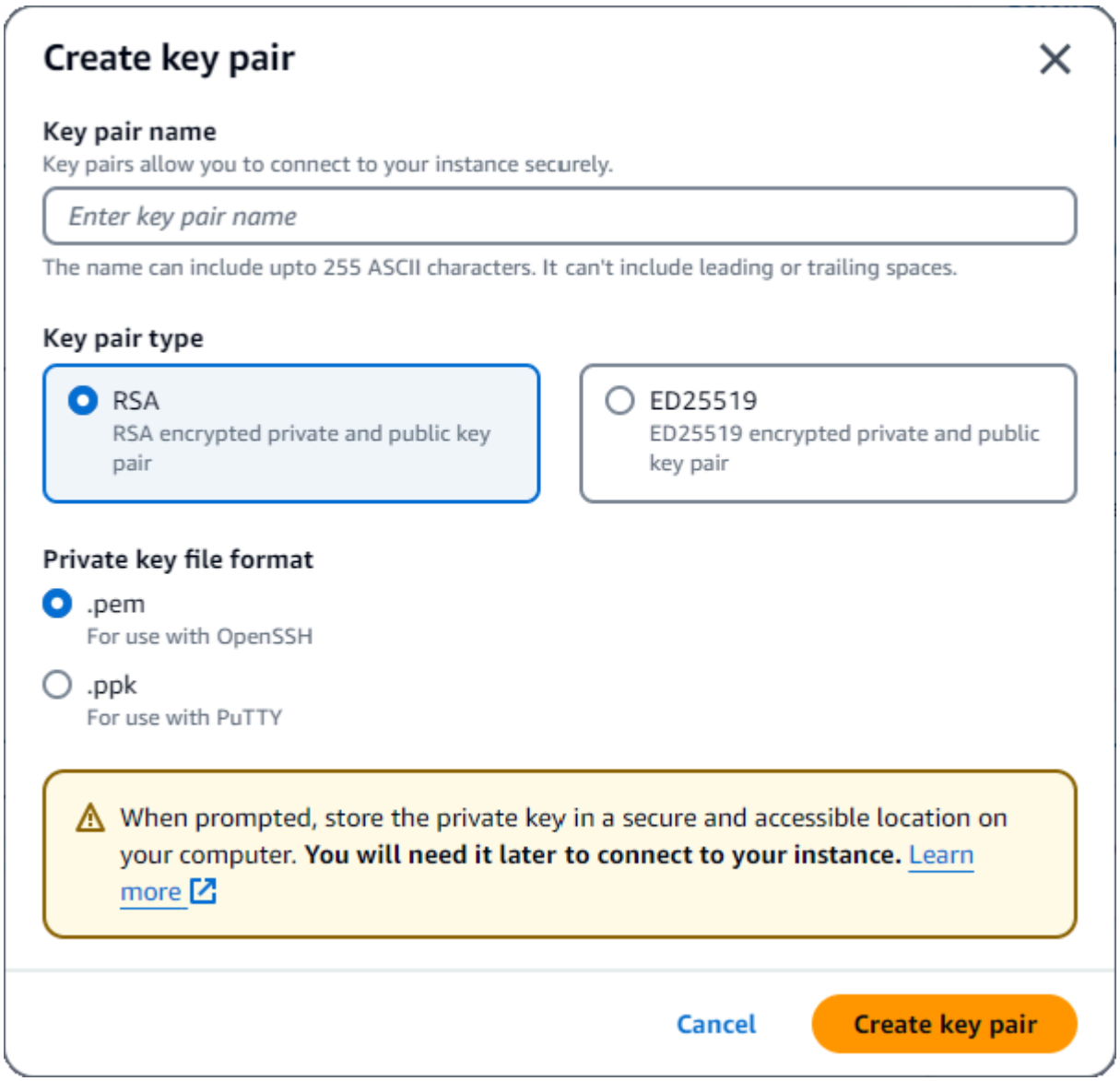
   Vous devez fournir une paire de clés Amazon EC2.

   Si vous possédez une paire de clés Amazon EC2 :

   a. Sélectionnez une paire de clés, choisissez votre paire de clés dans la liste.
   b. Vous devez déjà disposer du fichier de clé privée (fichier .pem ou .ppk) pour vous connecter à votre instance Amazon EC2.

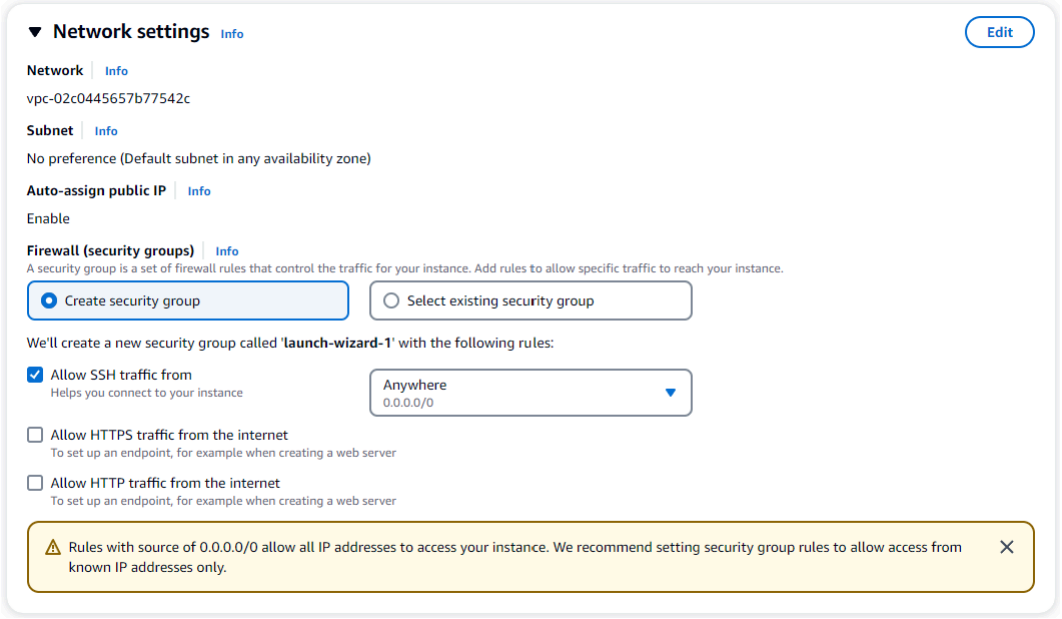
   Si vous ne possédez pas de paire de clés Amazon EC2 :

   a. Choisissez Créer une nouvelle paire de clés, la boîte de dialogue Créer une paire de clés apparaît.
   b. Entrez un nom dans le champ Nom de la paire de clés.
   c. Choisissez le type de paire de clés et le format de fichier de clé privée.
   d. Choisissez Créer une paire de clés.



> **Note**
>
> Pour des raisons de sécurité, nous vous recommandons vivement d'utiliser une paire de clés pour la connexion SSH et Internet à votre instance EC2.

6. Dans la section Paramètres réseau, sous Pare-feu (groupes de sécurité), choisissez Créer un groupe de sécurité ou Sélectionner un groupe de sécurité existant.



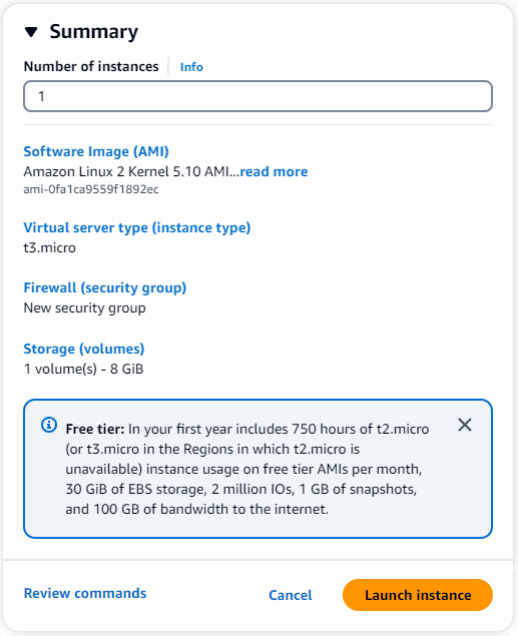
Si vous avez choisi de sélectionner un groupe de sécurité existant, sélectionnez-en un dans la liste déroulante Groupes de sécurité communs.

Si vous avez choisi de créer un nouveau groupe de sécurité, effectuez les opérations suivantes :

a. Vérifiez toutes les règles d'autorisation de trafic qui s'appliquent à votre connectivité EC2.

b. Dans le champ IP, choisissez Mon adresse IP ou sélectionnez Personnalisé pour choisir parmi une liste de blocs CIDR, de listes de préfixes ou de groupes de sécurité. Nous ne recommandons pas le choix Anywhere, sauf si votre instance EC2 se trouve sur un réseau isolé, car cela permet à n'importe quelle adresse IP d'accéder à votre instance EC2.

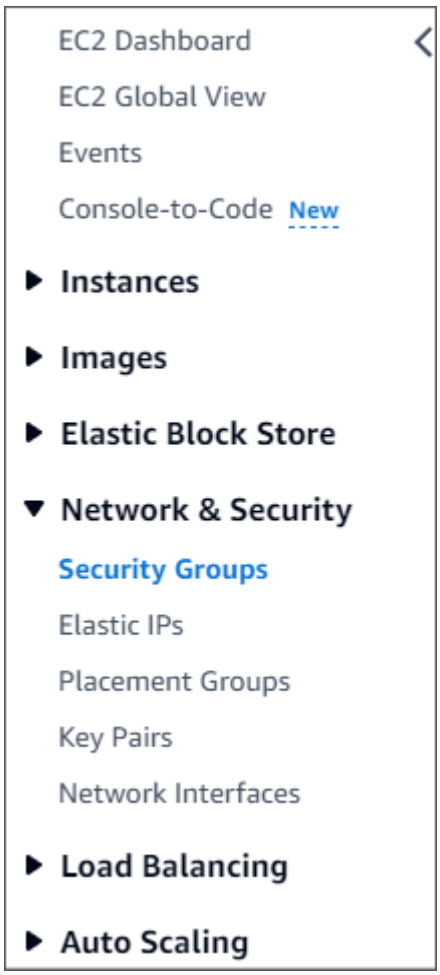
7. Dans la section Résumé, passez en revue votre configuration EC2 et choisissez Launch instance si c'est correct. Modifiez les groupes de sécurité.



Étape 2 : Créer un groupe de sécurité

Vous allez maintenant créer un nouveau groupe de sécurité dans votre Amazon VPC par défaut. Le groupe de sécurité vous `demoDocDB` permet de vous connecter à votre cluster Amazon DocumentDB sur le port 27017 (le port par défaut pour Amazon DocumentDB) depuis votre instance Amazon EC2.

1. Sur la console de gestion Amazon EC2, sous Réseau et sécurité, sélectionnez Groupes de sécurité.



2. Sélectionnez Create security group (Créer un groupe de sécurité).

3. Dans la section Informations de base :

   a. Sous Security group name (Nom du groupe de sécurité), saisissez `demoDocDB`.

   b. Pour Description, entrez une description.

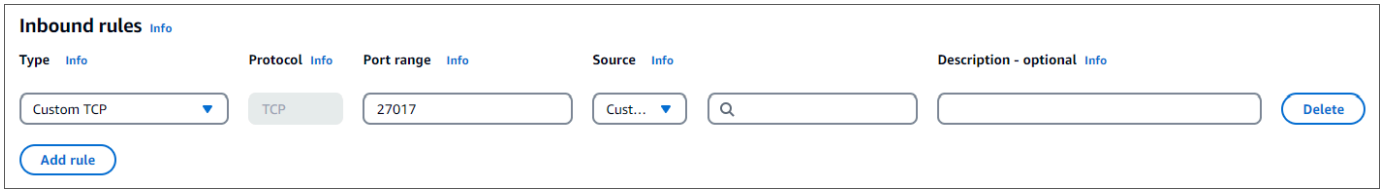
   c. Pour le VPC, acceptez l'utilisation de votre VPC par défaut.

4. Dans la section Règles entrantes, choisissez Ajouter une règle.

   a. Pour Type, choisissez Règle TCP personnalisée.

   b. Dans Portée de ports, entrez`27017`.

   c. Pour Destination, choisissez Personnalisé. Dans le champ à côté, recherchez le groupe de sécurité que vous venez d'appeler`demoEC2`. Vous devrez peut-être actualiser votre navigateur pour que la console Amazon EC2 renseigne automatiquement le nom de la source. `demoEC2`

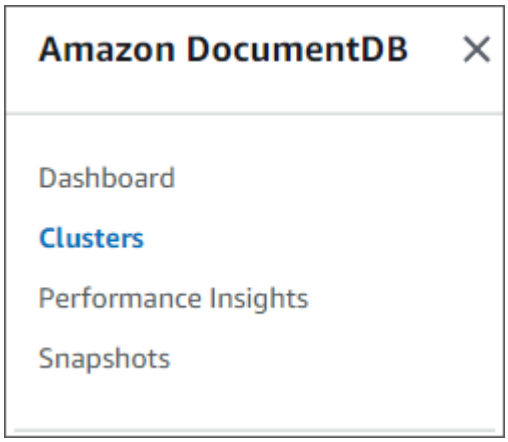


5. Acceptez toutes les autres valeurs par défaut et choisissez Create security group.

Étape 3 : créer un cluster Amazon DocumentDB

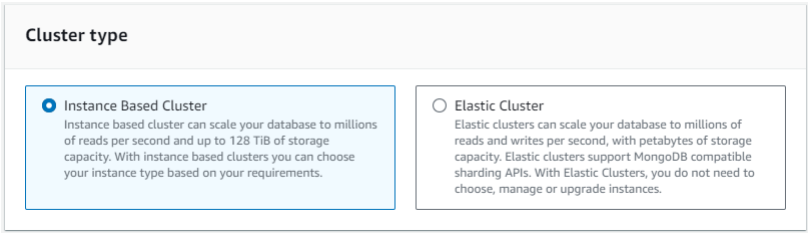
Pendant le provisionnement de l'instance Amazon EC2, vous allez créer votre cluster Amazon DocumentDB.

1. Accédez à la console Amazon DocumentDB et choisissez Clusters dans le volet de navigation.

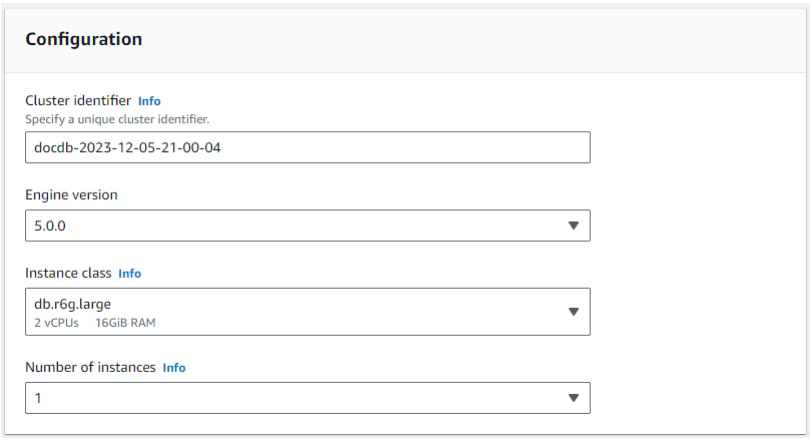


2. Choisissez Créer.

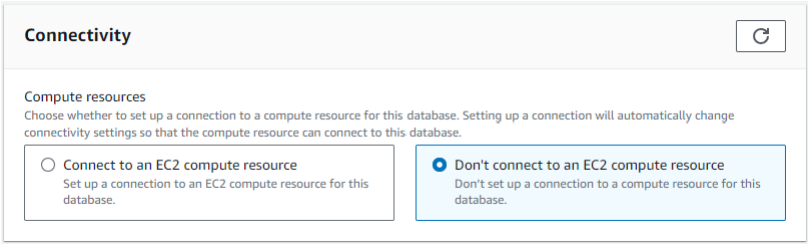
3. Conservez le paramètre Type de cluster à sa valeur par défaut, Cluster basé sur une instance.



4. Pour Nombre d'instances, choisissez 1. Cela minimisera les coûts. Conservez les autres paramètres par défaut.

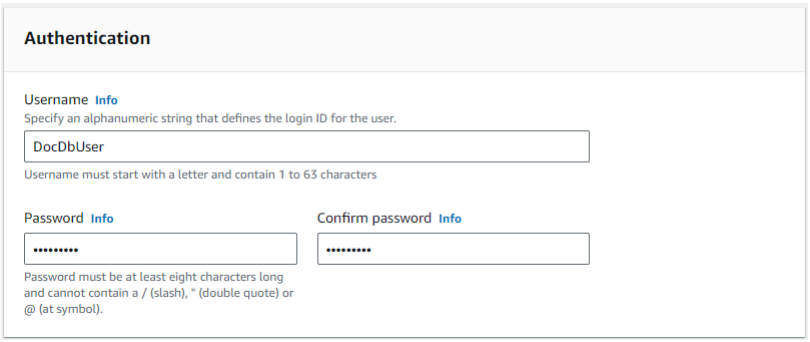


5. Pour Connectivité, conservez le paramètre par défaut Ne pas se connecter à une ressource de calcul EC2.

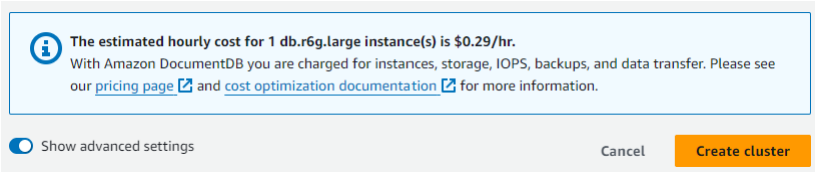


> **Note**
>
> La connexion à une ressource de calcul EC2 crée automatiquement des groupes de sécurité pour la connexion de votre ressource de calcul EC2 à votre cluster. Comme vous avez créé manuellement ces groupes de sécurité à l'étape précédente, vous devez sélectionner Ne pas vous connecter à une ressource de calcul EC2 afin de ne pas créer un deuxième ensemble de groupes de sécurité.

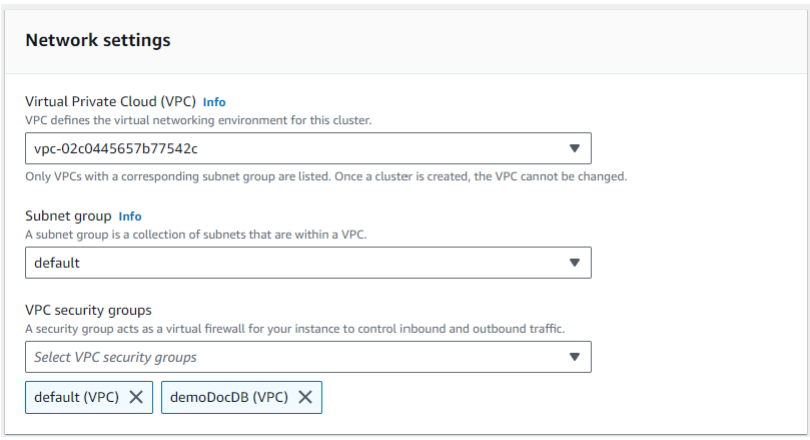
6. Pour l'authentification, entrez les informations de connexion. Important : Vous aurez besoin des informations de connexion pour authentifier votre cluster ultérieurement.



7. Activez Afficher les paramètres avancés.



8. Dans la section Paramètres réseau, pour les groupes de sécurité Amazon VPC, choisissez DemoDocDB.

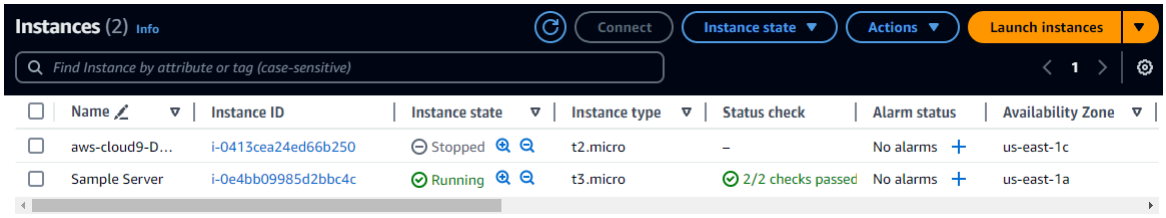


9. Choisissez Créer un cluster.

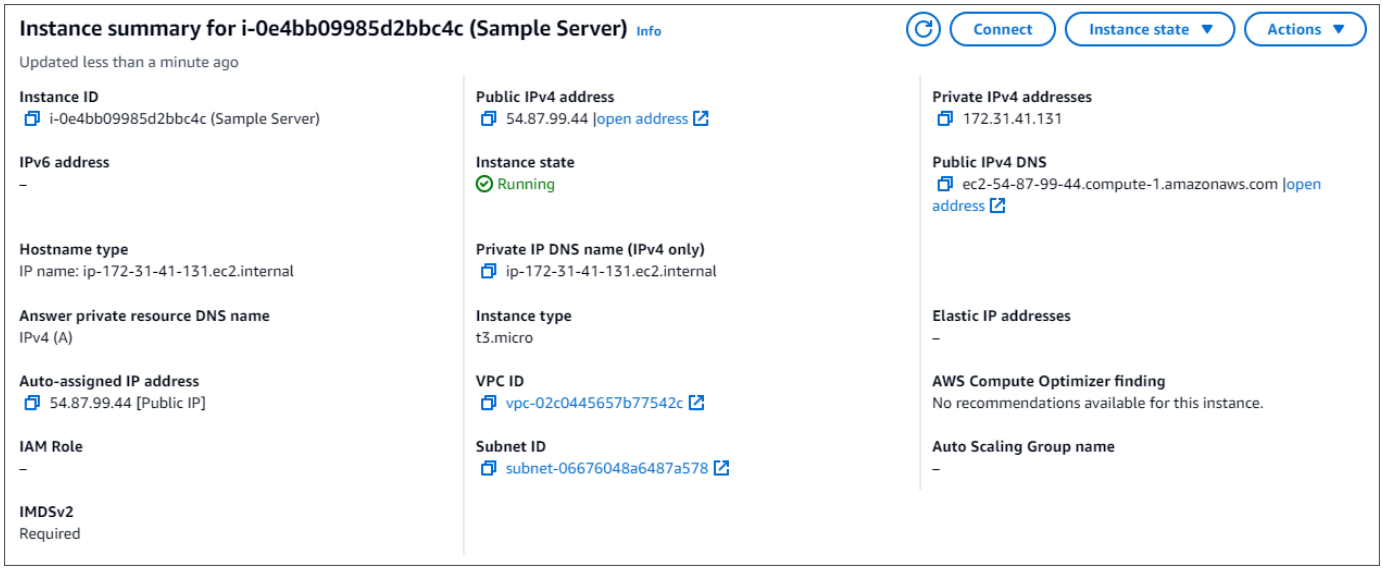
Étape 4 : Connectez-vous à votre instance Amazon EC2

Pour installer le shell mongo, vous devez d'abord vous connecter à votre instance Amazon EC2. L'installation du shell mongo vous permet de vous connecter à votre cluster Amazon DocumentDB et de l'interroger. Procédez comme suit :

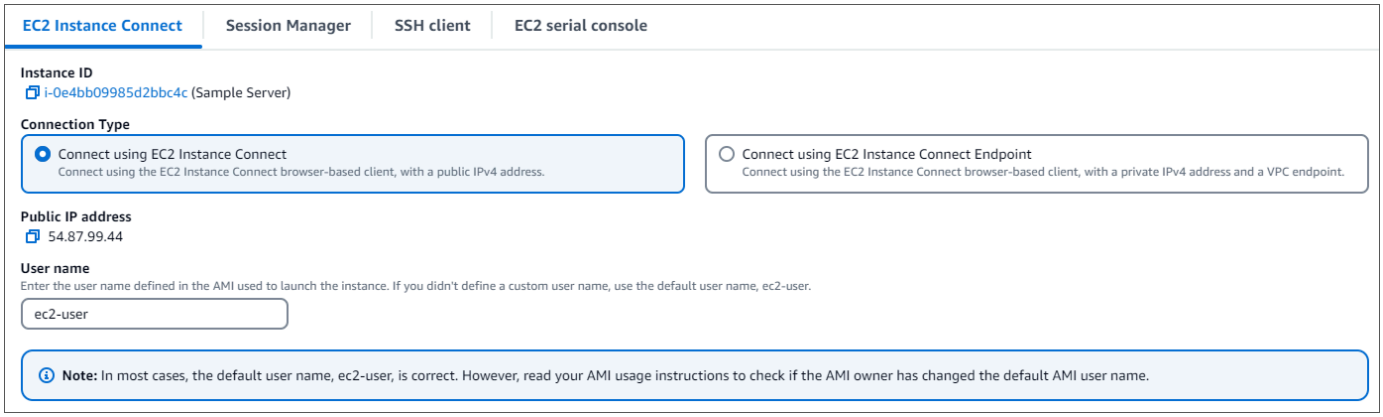
1. Sur la console Amazon EC2, accédez à vos instances et vérifiez si l'instance que vous venez de créer est en cours d'exécution. Si tel est le cas, sélectionnez l'instance en cliquant sur son ID.



2. Choisissez Se connecter.



3. Il existe quatre options à onglets pour votre méthode de connexion : Amazon EC2 Instance Connect, Session Manager, client SSH ou console série EC2. Vous devez en choisir un et suivre les instructions. Lorsque vous avez terminé, choisissez Connect.



> **Note**
>
> Si votre adresse IP a changé après le début de cette procédure, ou si vous revenez dans votre environnement ultérieurement, vous devez mettre à jour la règle de trafic entrant de votre groupe de `demoEC2` sécurité pour activer le trafic entrant depuis votre nouvelle adresse API.

Étape 5 : Installation du shell Mongo

Vous pouvez désormais installer le shell mongo, qui est un utilitaire de ligne de commande que vous utilisez pour connecter et interroger votre cluster Amazon DocumentDB. Suivez les instructions ci-dessous pour installer le shell mongo pour votre système d'exploitation.

On Amazon Linux

Pour installer le shell mongo sur Amazon Linux

1. Créez le fichier du référentiel. Sur la ligne de commande de votre instance EC2, exécutez la commande suivante :

```
echo -e "[mongodb-org-5.0] \nname=MongoDB Repository\nbaseurl=https://
repo.mongodb.org/yum/amazon/2/mongodb-org/5.0/x86_64/\ngpgcheck=1 \nenabled=1
 \ngpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc" | sudo tee /etc/
yum.repos.d/mongodb-org-5.0.repo
```

2. Une fois l'opération terminée, installez le shell mongo en exécutant la commande suivante :

```
sudo yum install -y mongodb-org-shell
```

On Ubuntu 18.04

Pour installer le shell mongo sur Ubuntu 18.04

1. Importez la clé publique qui sera utilisé par le système de gestion des packages.

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv
 2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5
```

2. Créez le fichier de liste `/etc/apt/sources.list.d/mongodb-org-3.6.list` pour MongoDB en utilisant la commande appropriée pour votre version d'Ubuntu.

   Ubuntu 18.04

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu xenial/
mongodb-org/3.6 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-
org-3.6.list
```

> **Note**
>
> La commande ci-dessus installera le shell mongo 3.6 pour Bionic et Xenial.

3. Recharger la base de données de package locale à l'aide de la commande suivante :

```
sudo apt-get update
```

4. Installez le shell MongoDB.

```
sudo apt-get install -y mongodb-org-shell
```

Pour plus d'informations sur l'installation de versions antérieures de MongoDB sur votre système Ubuntu, consultez Installation de MongoDB Community Edition sur Ubuntu

On other operating systems

Pour installer le shell mongo sur d'autres systèmes d'exploitation, consultez les informations relatives à l'installation de MongoDB Community Edition, dans la documentation MongoDB.

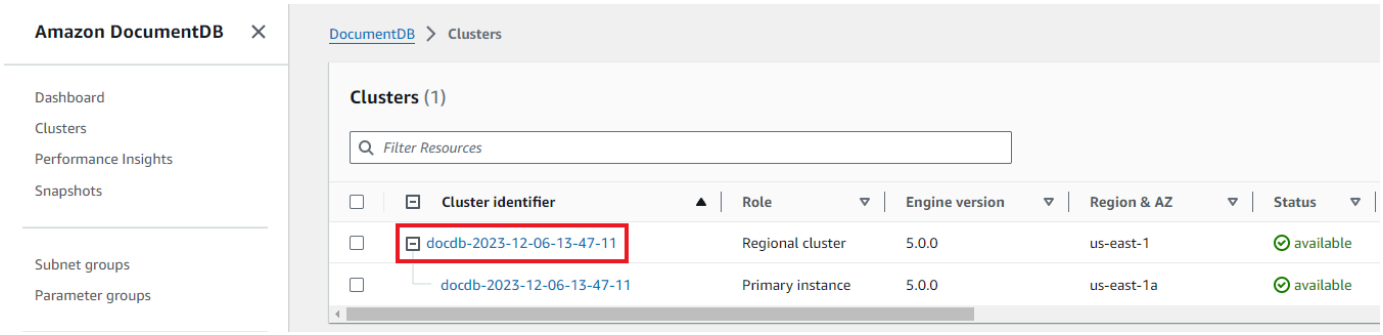
Étape 6 : Gérer le protocole TLS d'Amazon DocumentDB

Téléchargez le certificat CA pour Amazon DocumentDB avec le code suivant : `wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem`

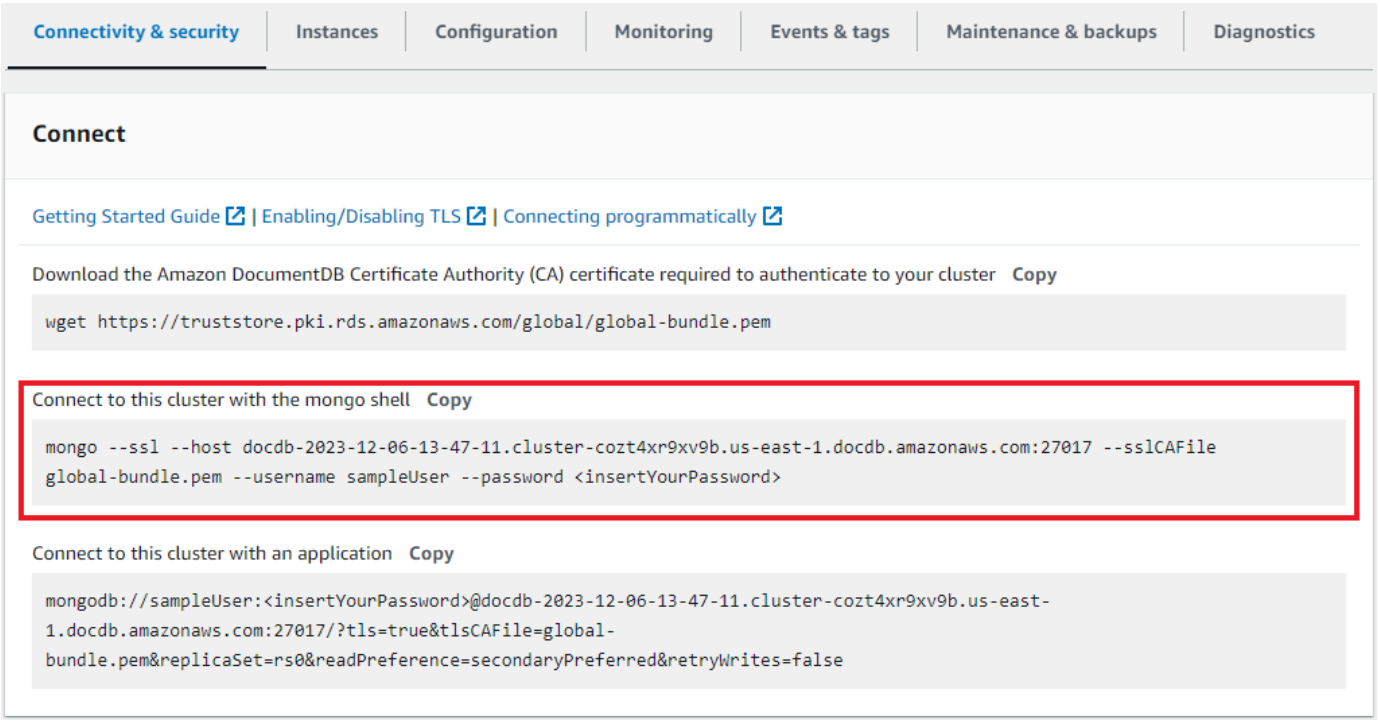
> **Note**
>
> Le protocole TLS (Transport Layer Security) est activé par défaut pour tous les nouveaux clusters Amazon DocumentDB. Pour plus d'informations, consultez Gestion des paramètres TLS du cluster Amazon DocumentDB.

Étape 7 : Connectez-vous à votre cluster Amazon DocumentDB

1. Sur la console Amazon DocumentDB, sous Clusters, localisez votre cluster. Choisissez le cluster que vous avez créé en cliquant sur l'identifiant du cluster.



2. Dans l'onglet Connectivité et sécurité, recherchez Connect to this cluster with the mongo shell dans le champ Connect :



Copiez la chaîne de connexion fournie et collez-la dans votre terminal.

Apportez-y les modifications suivantes :

a. Assurez-vous d'avoir le nom d'utilisateur correct dans la chaîne.

b. Omettez `<insertYourPassword>` afin que le shell mongo vous demande le mot de passe lorsque vous vous connectez.

Votre chaîne de connexion doit ressembler à ce qui suit :

```
mongo --ssl host docdb-2020-02-08-14-15-11.
cluster.region.docdb.amazonaws.com:27107 --sslCAFile global-bundle.pem
--username demoUser --password
```

3. Appuyez sur Entrée dans votre terminal. Vous êtes maintenant invité à saisir votre mot de passe. Entrez votre mot de passe.
4. Lorsque vous entrez votre mot de passe et que l'`rs0:PRIMARY>`invite s'affiche, vous êtes connecté avec succès à votre cluster Amazon DocumentDB.

Vous rencontrez des problèmes de connexion ? Consultez la section Résolution des problèmes liés à Amazon DocumentDB.

Étape 8 : Insérer et interroger des données

Maintenant que vous êtes connecté à votre cluster, vous pouvez exécuter quelques requêtes pour vous familiariser avec l'utilisation d'une base de données de documents.

1. Pour insérer un seul document, entrez les informations suivantes :

```
db.collection.insert({"hello":"DocumentDB"})
```

2. Vous obtenez le résultat suivant :

```
WriteResult({ "nInserted" : 1 })
```

3. Vous pouvez lire le document que vous avez écrit avec la `findOne()` commande (car il ne renvoie qu'un seul document). Entrez les informations suivantes :

```
db.collection.findOne()
```

4. Vous obtenez le résultat suivant :

```
{ "_id" : ObjectId("5e401fe56056fda7321fbd67"), "hello" :
"DocumentDB" }
```

5. Pour effectuer quelques requêtes supplémentaires, considérez un cas d'utilisation de profils de jeu. Tout d'abord, insérez quelques entrées dans une collection intituléeprofiles. Entrez les informations suivantes :

```
db.profiles.insertMany([
```

```
           { "_id" : 1, "name" : "Matt", "status": "active", "level": 12,
"score":202},
           { "_id" : 2, "name" : "Frank", "status": "inactive", "level": 2,
"score":9},
           { "_id" : 3, "name" : "Karen", "status": "active", "level": 7,
"score":87},
           { "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
"score":27}
           ])
```

6. Vous obtenez le résultat suivant :

```
{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4 ] }
```

7. Utilisez la `find()` commande pour renvoyer tous les documents de la collection de profils. Entrez les informations suivantes :

```
db.profiles.find()
```

8. Vous obtiendrez une sortie qui correspondra aux données que vous avez saisies à l'étape 5.
9. Utilisez une requête pour un seul document à l'aide d'un filtre. Entrez les informations suivantes :

```
db.profiles.find({name: "Katie"})
```

10. Vous devriez récupérer cette sortie :

```
{ "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
"score":27}
```

11. Essayons maintenant de trouver un profil et de le modifier à l'aide de la `findAndModify` commande. Nous allons donner dix points supplémentaires à l'utilisateur Matt avec le code suivant :

```
db.profiles.findAndModify({
    query: { name: "Matt", status: "active"},
    update: { $inc: { score: 10 } }
})
```

12. Vous obtenez le résultat suivant (notez que son score n'a pas encore augmenté) :

```
{
"_id" : 1,
"name" : "Matt",
"status" : "active",
"level" : 12,
"score" : 202
}
```

13. Vous pouvez vérifier que son score a changé avec la requête suivante :

```
db.profiles.find({name: "Matt"})
```

14. Vous obtenez le résultat suivant :

```
{ "_id" : 1, "name" : "Matt", "status" : "active", "level" : 12,
"score" : 212 }
```

Étape 9 : Explorez

Félicitations ! Vous avez terminé avec succès le guide de démarrage rapide d'Amazon DocumentDB.

Quelle est la prochaine étape ? Découvrez comment tirer pleinement parti de cette puissante base de données avec certaines de ses fonctionnalités populaires :

- Gestion d'Amazon DocumentDB
- Dimensionnement
- Sauvegarde et restauration

> **Note**
>
> Pour réduire les coûts, vous pouvez soit arrêter votre cluster Amazon DocumentDB afin de réduire les coûts, soit supprimer le cluster. Par défaut, après 30 minutes d'inactivité, votre AWS Cloud9 environnement arrête l'instance Amazon EC2 sous-jacente.

Connectez-vous à l'aide du pilote Amazon DocumentDB JDBC

Le pilote JDBC pour Amazon DocumentDB fournit une interface SQL relationnelle aux développeurs et permet la connectivité à partir d'outils de BI tels que Tableau et. DbVisualizer

Pour plus d'informations, consultez la documentation du pilote Amazon DocumentDB JDBC sur. GitHub

Rubriques

Premiers pas

Étape 1. Création d'un cluster Amazon DocumentDB

Si aucun cluster Amazon DocumentDB n'a été créé, créez-en un en suivant les instructions de la section Getting Started du manuel du développeur Amazon DocumentDB.

> **Note**
>
> DocumentDB est un service réservé au Virtual Private Cloud (VPC). Si vous vous connectez depuis une machine locale, en dehors du VPC du cluster, vous devez créer une connexion SSH vers une instance Amazon EC2. Dans ce cas, lancez votre cluster en suivant les instructions de Connect with EC2. Consultez Utiliser un tunnel SSH pour vous connecter à Amazon DocumentDB pour plus d'informations sur le tunneling SSH et savoir quand vous pourriez en avoir besoin.

Étape 2. Installation de JRE ou de JDK

En fonction de votre application BI, vous devrez peut-être vous assurer qu'une installation JRE ou JDK 64 bits version 8 ou ultérieure est installée sur votre ordinateur. Vous pouvez télécharger le Java SE Runtime Environment 8 ici.

Étape 3. Téléchargez le pilote JDBC DocumentDB

Téléchargez le pilote JDBC DocumentDB ici. Le pilote est empaqueté sous la forme d'un seul fichier JAR (par exemple documentdb-jdbc-1.0.0-all.jar).

Étape 4 : Utilisation d'un tunnel SSH pour se connecter à Amazon DocumentDB

Les clusters Amazon DocumentDB (compatibles avec MongoDB) sont déployés au sein d'un Amazon Virtual Private Cloud (Amazon VPC). Ils sont accessibles directement par les instances Amazon EC2 ou d'autres AWS services déployés dans le même Amazon VPC. En outre, Amazon DocumentDB est accessible par des instances EC2a ou d'autres AWS services dans différents VPC de la même AWS région ou dans d'autres régions via le peering VPC.

Vous pouvez utiliser le tunneling SSH (également appelé redirection de port) pour accéder à vos ressources Amazon DocumentDB, depuis l'extérieur du VPC du cluster. Ce sera le cas pour la plupart des utilisateurs qui n'exécutent pas leur application sur une machine virtuelle dans le même VPC que le cluster DocumentDB.

Pour créer un tunnel SSH, vous avez besoin d'une instance Amazon EC2 exécutée dans le même Amazon VPC que votre cluster Amazon DocumentDB. Vous pouvez soit utiliser une instance EC2 existante dans le même VPC que votre cluster, soit en créer une. Vous pouvez configurer un tunnel SSH vers le `sample-cluster.node.us-east-1.docdb.amazonaws.com` cluster Amazon DocumentDB en exécutant la commande suivante sur votre ordinateur local.

```
ssh -i "ec2Access.pem" -L 27017:sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017 ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com -
N
```

L'indicateur -L est utilisé pour transférer un port local. Il s'agit d'une condition préalable à la connexion à tout outil de BI exécuté sur un client extérieur à votre VPC. Une fois que vous avez exécuté l'étape ci-dessus, vous pouvez passer aux étapes suivantes pour l'outil de BI de votre choix.

Pour plus d'informations sur le tunneling SSH, consultez la documentation sur l'utilisation d'un tunnel SSH pour vous connecter à Amazon DocumentDB.

Connectez-vous à Amazon DocumentDB depuis Tableau Desktop

Rubriques

- Ajout du pilote JDBC Amazon DocumentDB
- Connexion à Amazon DocumentDB à l'aide de Tableau - Tunnel SSH

Ajout du pilote JDBC Amazon DocumentDB

Pour vous connecter à Amazon DocumentDB depuis Tableau Desktop, vous devez télécharger et installer le pilote JDBC DocumentDB et le connecteur DocumentDB Tableau.

1. Téléchargez le fichier JAR du pilote JDBC DocumentDB et copiez-le dans l'un des répertoires suivants en fonction de votre système d'exploitation :
   - Fenêtres - `C:\Program Files\Tableau\Drivers`
   - macOS - `~/Library/Tableau/Drivers`
2. Téléchargez le connecteur DocumentDB Tableau (un fichier TACO) et copiez-le dans votre répertoire My Tableau Repository/Connectors.
   - Fenêtres - `C:\Users\[user]\Documents\My Tableau Repository\Connectors`
   - macOS - `/Users/[user]/Documents/My Tableau Repository/Connectors`

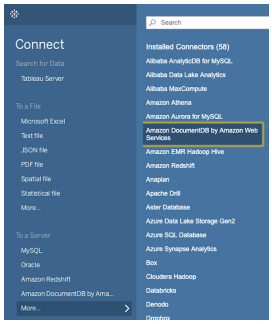
Pour plus d'informations, consultez la documentation Tableau.

> **Note**
>
> Si vous utilisez des certificats CA plus récents, assurez-vous de mettre à niveau votre pilote JDBC vers la version v1.4.5 (disponible dans ce AWS GitHub référentiel).

Connexion à Amazon DocumentDB à l'aide de Tableau - Tunnel SSH

Pour vous connecter à Tableau depuis une machine cliente extérieure au VPC de votre cluster DocumentDB, vous devez configurer un tunnel SSH avant de suivre les étapes ci-dessous :

1. Lancez l'application Tableau Desktop.
2. Accédez à Connect > To A Server > Plus.
3. Choisissez Amazon DocumentDB d'Amazon Web Services sous Connecteurs installés.

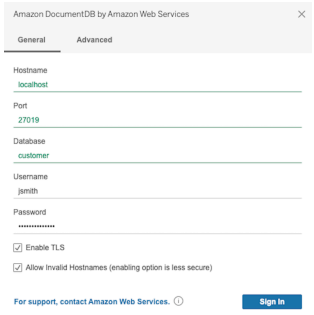


Connexion à Amazon DocumentDB à l'aide de Tableau - Tunnel SSH externe

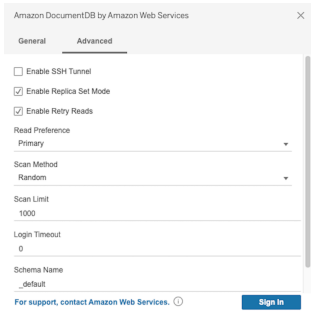
1. Entrez les paramètres de connexion requis : nom d'hôte, port, base de données, nom d'utilisateur et mot de passe. Les paramètres de connexion de l'exemple ci-dessous sont équivalents à la chaîne de connexion JDBC :

   ```
   jdbc:documentdb://localhost:27019/test?
   tls=true&tlsAllowInvalidHostnames=true&scanMethod=random&scanLimit=1000&logir
   ```

   les paramètres de nom d'utilisateur et de mot de passe transmis séparément dans une collection de propriétés. Pour plus d'informations sur les paramètres des chaînes de connexion, consultez la documentation github du pilote Amazon DocumentDB JDBC.



2. (Facultatif) Des options plus avancées se trouvent dans l'onglet Avancé.

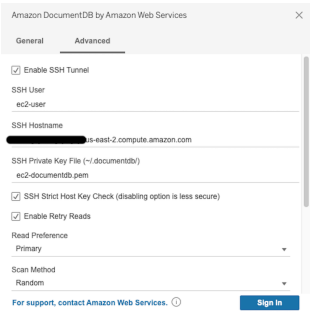


3. Choisissez Sign in (Connexion).

Connexion à Amazon DocumentDB à l'aide de Tableau - Tunnel SSH interne

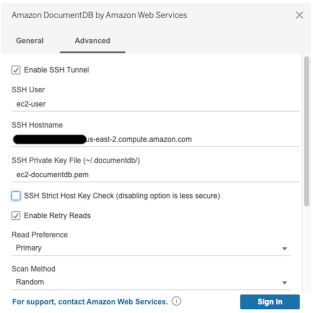
> **Note**
>
> Si vous préférez ne pas configurer le tunnel SSH à l'aide d'un terminal, vous pouvez utiliser l'interface graphique de Tableau pour spécifier les détails de votre instance EC2, que le pilote JDBC utilisera de manière inhérente pour créer un tunnel SSH.

1. Dans l'onglet Avancé, choisissez l'option Activer le tunnel SSH pour consulter d'autres propriétés.



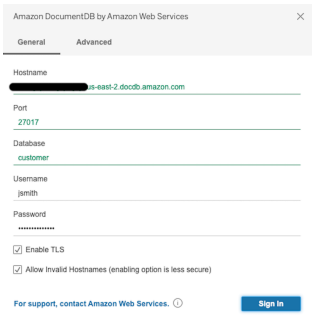
2. Entrez l'utilisateur SSH, le nom d'hôte SSH et le fichier de clé privée SSH.
3. (Facultatif) Vous pouvez désactiver l'option SSH Strict Host Key Check qui contourne la vérification de la clé d'hôte par rapport à un fichier d'hôtes connu.

   > **Note**
   >
   > La désactivation de cette option est moins sûre car elle peut entraîner une man-in-the-middleattaque.



4. Entrez les paramètres requis : nom d'hôte, port, base de données, nom d'utilisateur et mot de passe.

> **Note**
>
> Assurez-vous d'utiliser le point de terminaison du cluster DocumentDB et non localhost lorsque vous utilisez l'option de tunnel SSH interne.



5. Choisissez Sign in (Connexion).

Connectez-vous à Amazon DocumentDB depuis DbVisualizer

Rubriques

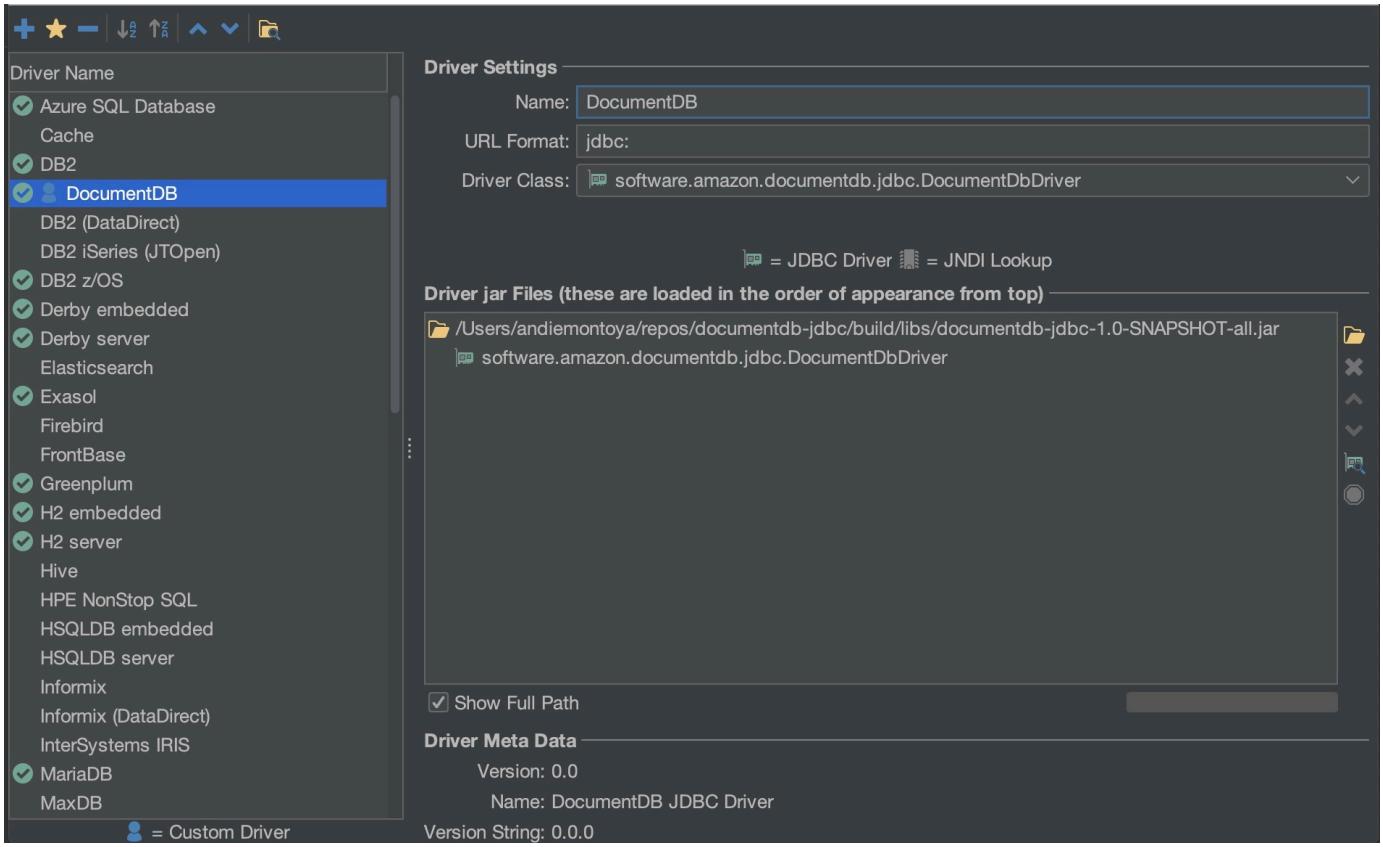
- Ajout du pilote JDBC Amazon DocumentDB
- Connexion à Amazon DocumentDB à l'aide de DbVisualizer

Ajout du pilote JDBC Amazon DocumentDB

Pour vous connecter à Amazon DocumentDB depuis, DbVisualizer vous devez d'abord importer le pilote Amazon DocumentDB JDBC

1. Démarrez l' DbVisualizer application et accédez au chemin du menu : Outils > Gestionnaire de pilotes...
2. Choisissez + (ou dans le menu, sélectionnez Pilote > Créer un pilote).
3. Définissez Nom sur `DocumentDB`.
4. Définissez le format de l'URL sur `jdbc:documentdb://<host>[:port]/<database>[?option=value[&option=value[...]]]`
5. Cliquez sur le bouton du dossier, puis sélectionnez le fichier JAR du pilote JDBC Amazon DocumentDB et cliquez sur le bouton Ouvrir.

6. Vérifiez que le champ Driver Class est défini sur`software.amazon.documentdb.jdbc.DocumentDbDriver`. Les paramètres de votre gestionnaire de pilotes pour DocumentDB doivent ressembler à l'exemple suivant.



7. Fermez la boîte de dialogue. Le pilote JDBC Amazon DocumentDB sera configuré et prêt à être utilisé.

Connexion à Amazon DocumentDB à l'aide de DbVisualizer

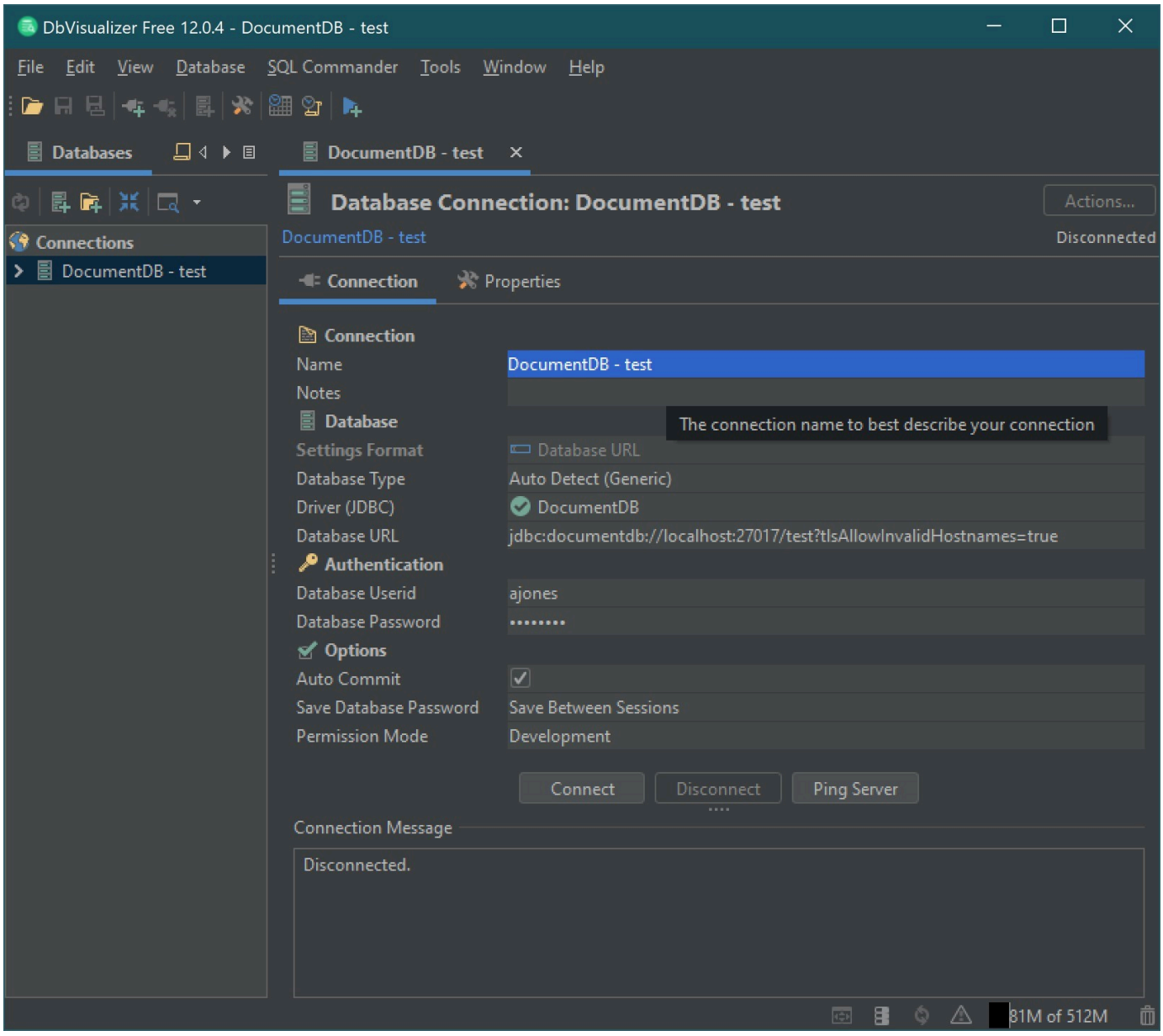
Connectez-vous à Amazon DocumentDB à l'aide de DbVisualizer

1. Si vous vous connectez depuis l'extérieur du VPC du cluster Amazon DocumentDB, assurez-vous d'avoir configuré un tunnel SSH.
2. Choisissez Base de données > Créer une connexion à la base de données dans le menu supérieur.
3. Entrez un nom descriptif pour le champ Nom.
4. Définissez Driver (JDBC) sur le pilote DocumentDB que vous avez créé dans la section précédente.
5. Définissez l'URL de la base de données sur votre chaîne de connexion JDBC.

Par exemple : `jdbc:documentdb://localhost:27017/database?tlsAllowInvalidHostnames=true`

6. Définissez Database Userid sur votre ID utilisateur Amazon DocumentDB.
7. Définissez le mot de passe de la base de données sur le mot de passe correspondant à l'ID utilisateur.

   Votre boîte de dialogue de connexion à la base de données doit ressembler à la boîte de dialogue suivante :



8. Choisissez Se connecter.

Génération automatique de schémas JDBC

Amazon DocumentDB est une base de données de documents et ne possède donc pas le concept de tables et de schémas. Cependant, les outils de BI tels que Tableau s'attendent à ce que la base de données qu'ils connectent présente un schéma. Plus précisément, lorsque la connexion au pilote JDBC doit obtenir le schéma de la collection dans la base de données, elle interroge toutes les collections de la base de données. Le pilote déterminera si une version mise en cache du schéma pour cette collection existe déjà. Si aucune version mise en cache n'existe, elle échantillonne la collection pour les documents et crée un schéma basé sur le comportement suivant.

Rubriques

Limites de génération de schémas

Le pilote JDBC DocumentDB limite la longueur des identifiants à 128 caractères. Le générateur de schéma peut tronquer la longueur des identifiants générés (noms de tables et noms de colonnes) pour s'assurer qu'ils correspondent à cette limite.

Options de méthode de numérisation

Le comportement d'échantillonnage peut être modifié à l'aide des options de chaîne de connexion ou de source de données.

- Méthode de scannage= <option>
  - random - (par défaut) - Les exemples de documents sont renvoyés dans un ordre aléatoire.
  - IDForward - Les exemples de documents sont renvoyés par ordre d'identification.
  - IDReverse - Les exemples de documents sont renvoyés dans l'ordre inverse de l'identifiant.
  - tous - Échantillonnez tous les documents de la collection.
- ScanLimit= <n>- Le nombre de documents à échantillonner. La valeur doit être un nombre entier positif. La valeur par défaut est 1000. Si ScanMethod est défini sur all, cette option est ignorée.

Types de données Amazon DocumentDB

Le serveur DocumentDB prend en charge un certain nombre de types de données MongoDB. Les types de données pris en charge et les types de données JDBC associés sont répertoriés ci-dessous.

Type de données MongoDB	Pris en charge dans DocumentDB	Type de données JDBC
Données binaires	Oui	VARBINARY
Booléen	Oui	BOOLEAN
Double	Oui	DOUBLE
Entier 32 bits	Oui	INTEGER
Entier 64 bits	Oui	BIGINT
Chaîne	Oui	VARCHAR
ObjectId	Oui	VARCHAR
Date	Oui	TIMESTAMP
Null	Oui	VARCHAR
Expression régulière	Oui	VARCHAR
Horodatage	Oui	VARCHAR
MinKey	Oui	VARCHAR
MaxKey	Oui	VARCHAR
Objet	Oui	table virtuelle
Tableau	Oui	table virtuelle
Decimal128	Non	DECIMAL
JavaScript	Non	VARCHAR

Type de données MongoDB	Pris en charge dans DocumentDB	Type de données JDBC
JavaScript (avec lunette)	Non	VARCHAR
Non défini	Non	VARCHAR
Symbol	Non	VARCHAR
DBPointer (4.0 et versions ultérieures)	Non	VARCHAR

Cartographie de champs de documents scalaires

Lors de la numérisation d'un échantillon de documents d'une collection, le pilote JDBC crée un ou plusieurs schémas pour représenter les échantillons de la collection. En général, un champ scalaire du document correspond à une colonne du schéma de table. Par exemple, dans une collection nommée team et dans un seul document`{ "_id" : "112233", "name" : "Alastair", "age": 25 }`, cela correspondrait au schéma :

Nom de la table	Nom de la colonne	Type de données	Clé
équipe	identifiant de l'équipe	VARCHAR	PK
équipe	name	VARCHAR	
équipe	age	INTEGER	

Promotion des conflits liés aux types de données

Lors de la numérisation des documents échantillonnés, il est possible que les types de données d'un champ ne soient pas cohérents d'un document à l'autre. Dans ce cas, le pilote JDBC fera passer le type de données JDBC à un type de données commun qui conviendra à tous les types de données des documents échantillonnés.

Par exemple :

```
{
```

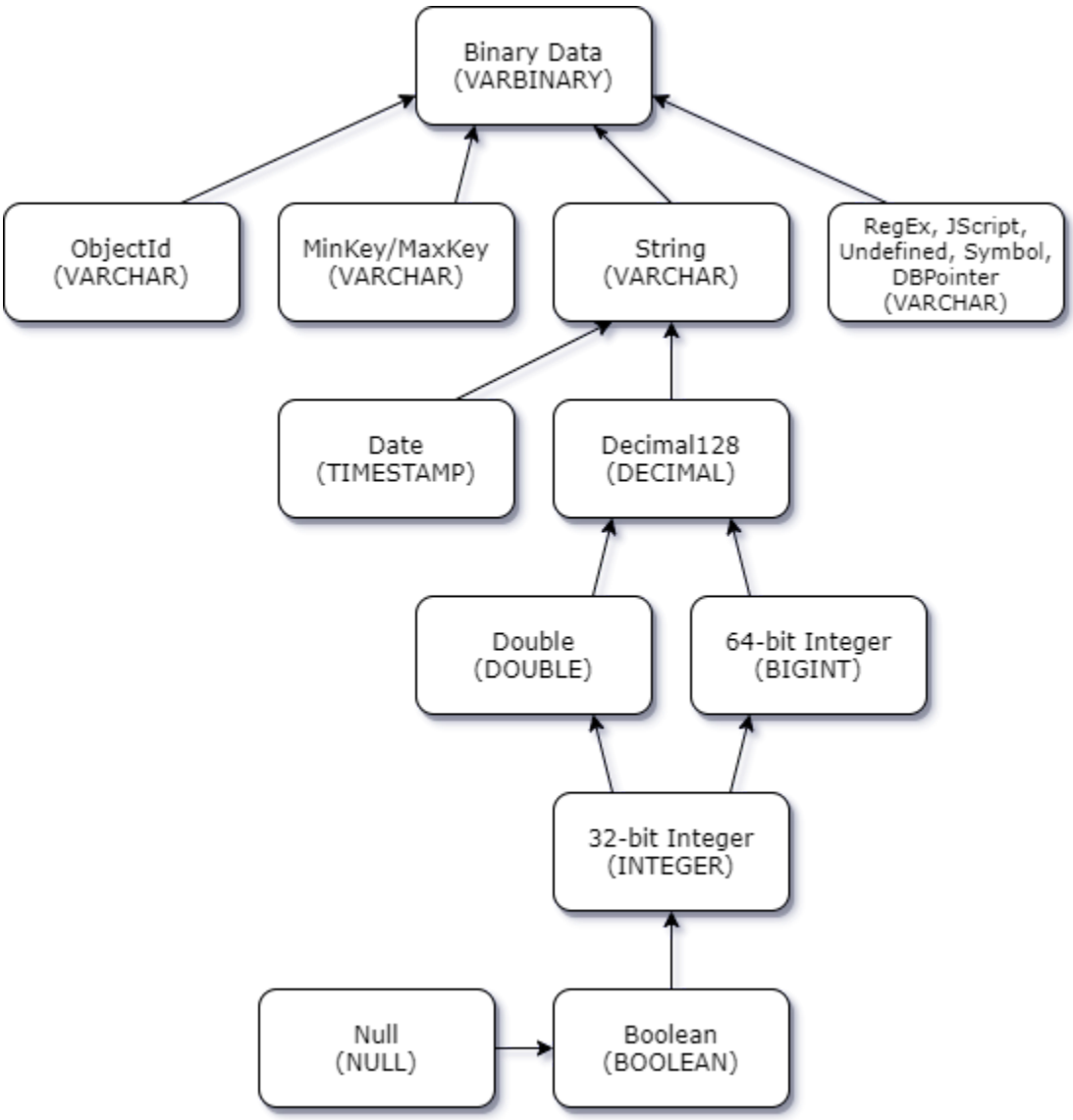
```
"_id" : "112233",
"name" : "Alastair", "age" : 25
}

{
"_id" : "112244",
"name" : "Benjamin",
"age" : "32"
}
```

Le champ d'âge est de type entier 32 bits dans le premier document mais de type chaîne dans le second document. Ici, le pilote JDBC fera passer le type de données JDBC à VARCHAR pour gérer l'un ou l'autre type de données lorsqu'il est rencontré.

Nom de la table	Nom de la colonne	Type de données	Clé
équipe	identifiant de l'équipe	VARCHAR	PK
équipe	name	VARCHAR	
équipe	age	VARCHAR	

Promotion des conflits scalaires

Le schéma suivant montre la manière dont les conflits de type de données scalaire-scalaire sont résolus.



Promotion des conflits de type complexe scalaire

À l'instar des conflits de type scalaire-scalaire, le même champ dans différents documents peut contenir des types de données contradictoires entre complexes (tableau et objet) et scalaires (entier, booléen, etc.). Tous ces conflits sont résolus (promus) à VARCHAR pour ces domaines. Dans ce cas, les données du tableau et de l'objet sont renvoyées sous forme de représentation JSON.

Exemple de conflit entre un tableau intégré et un champ de chaîne :

```
{
   "_id":"112233",
   "name":"George Jackson",
   "subscriptions":[
```

```
      "Vogue",
      "People",
      "USA Today"
   ]
}
{
   "_id":"112244",
   "name":"Joan Starr",
   "subscriptions":1
}
```

L'exemple ci-dessus correspond au schéma de la table customer2 :

Nom de la table	Nom de la colonne	Type de données	Clé
client 2	identifiant customer2	VARCHAR	PK
client 2	name	VARCHAR	
client 2	abonnement	VARCHAR	

et la table virtuelle customer1_subscription :

Nom de la table	Nom de la colonne	Type de données	Clé
customer1_abonnements	identifiant client1	VARCHAR	PK/FK
customer1_abonnements	subscriptions_index_lvl0	BIGINT	PK
customer1_abonnements	value	VARCHAR	
customer_address	city	VARCHAR	
customer_address	region	VARCHAR	
customer_address	country	VARCHAR	

Nom de la table	Nom de la colonne	Type de données	Clé
customer_address	code	VARCHAR	

Gestion des types de données d'objets et de tableaux

Jusqu'à présent, nous avons uniquement décrit la façon dont les types de données scalaires sont mappés. Les types de données Object et Array sont (actuellement) mappés à des tables virtuelles. Le pilote JDBC créera une table virtuelle pour représenter les champs d'un objet ou d'un tableau dans un document. Le nom de la table virtuelle mappée concaténera le nom de la collection d'origine suivi du nom du champ séparé par un trait de soulignement (« _ »).

La clé primaire de la table de base (« _id ») prend un nouveau nom dans la nouvelle table virtuelle et est fournie en tant que clé étrangère à la table de base associée.

Pour les champs de type tableau intégré, des colonnes d'index sont générées pour représenter l'index dans le tableau à chaque niveau du tableau.

Exemple de champ d'objet intégré

Pour les champs d'objet d'un document, un mappage vers une table virtuelle est créé par le pilote JDBC.

```
{
   "Collection: customer",
   "_id":"112233",
   "name":"George Jackson",
   "address":{
      "address1":"123 Avenue Way",
      "address2":"Apt. 5",
      "city":"Hollywood",
      "region":"California",
      "country":"USA",
      "code":"90210"
   }
}
```

L'exemple ci-dessus correspond au schéma de la table des clients :

Nom de la table	Nom de la colonne	Type de données	Clé
customer	identifiant du client	VARCHAR	PK
customer	name	VARCHAR	

et la table virtuelle customer_address :

Nom de la table	Nom de la colonne	Type de données	Clé
customer_address	identifiant du client	VARCHAR	PK/FK
customer_address	adresse1	VARCHAR	
customer_address	adresse2	VARCHAR	
customer_address	city	VARCHAR	
customer_address	region	VARCHAR	
customer_address	country	VARCHAR	
customer_address	code	VARCHAR	

Exemple de champ de tableau intégré

Pour les champs de tableau d'un document, un mappage vers une table virtuelle est également créé par le pilote JDBC.

```
{
   "Collection: customer1",
   "_id":"112233",
   "name":"George Jackson",
   "subscriptions":[
      "Vogue",
      "People",
      "USA Today"
   ]
}
```

L'exemple ci-dessus correspond au schéma de la table customer1 :

Nom de la table	Nom de la colonne	Type de données	Clé
client 1	identifiant client1	VARCHAR	PK
client 1	name	VARCHAR	

et la table virtuelle customer1_subscription :

Nom de la table	Nom de la colonne	Type de données	Clé
customer1_abonnements	identifiant client1	VARCHAR	PK/FK
customer1_abonnements	subscriptions_index_lvl0	BIGINT	PK
customer1_abonnements	value	VARCHAR	
customer_address	city	VARCHAR	
customer_address	region	VARCHAR	
customer_address	country	VARCHAR	
customer_address	code	VARCHAR	

Support et limites du SQL

Le pilote JDBC Amazon DocumentDB est un pilote en lecture seule qui prend en charge un sous-ensemble de SQL-92 et certaines extensions courantes. Reportez-vous à la documentation relative aux limitations SQL et à la documentation relative aux limitations JDBC pour plus d'informations.

Résolution des problèmes

[Si vous rencontrez des problèmes lors de l'utilisation du pilote Amazon DocumentDB JDBC, consultez le guide de résolution des problèmes.]

Connect à l'aide du pilote ODBC Amazon DocumentDB

Le pilote ODBC pour Amazon DocumentDB fournit une interface SQL relationnelle aux développeurs et permet la connectivité à partir d'outils de BI tels que Power BI Desktop et Microsoft Excel.

Pour plus d'informations, consultez la documentation du pilote ODBC Amazon DocumentDB sur GitHub.

Rubriques

Démarrer

Étape 1. Création de clusters Amazon DocumentDB

Si vous ne disposez pas déjà d'un Amazon DocumentDB Documentmentmentmentmentmentmentmentmentmentmentmentmentmentmentmentmentmentmentmen

> **Note**
>
> Amazon DocumentDB DocumentmentmentmentmentmentmentmentmentmentmentmentmentmentmentmentmentmentmentsClo Virtual Private Cloud (VPC). Si vous vous connectez depuis une machine locale en dehors du VPC du cluster, vous devez créer une connexion SSH à une instance Amazon EC2.

Dans ce cas, lancez votre cluster en suivant les instructions de la section Connect à EC2. Consultez la section Utilisation d'un tunnel SSH pour Connect à Amazon DocumentDB pour plus d'informations sur le tunneling SSH et les situations dans lesquelles vous pourriez en avoir besoin.

Étape 2. Installation d'un JRE ou d'un JDK

En fonction de votre application de BI, vous devrez peut-être vous assurer qu'une installation JRE ou JDK 64 bits version 8 ou ultérieure est installée sur votre ordinateur. Vous pouvez télécharger l'environnement d'exécution Java SE 8 ici.

Étape 3 Téléchargez le pilote ODBC Amazon DocumentDB

Téléchargez le pilote ODBC Amazon DocumentDB ici. Choisissez le programme d'installation approprié (par exemple, documentdb-odbc-1.0.0.msi). Suivez le guide d'installation.

Étape 4. Utilisation d'un tunnel SSH pour Connect à Amazon DocumentDB

Les clusters Amazon DocumentDB Documentmentmentmentmentmentmentmentmentmentmentmentmentment dans un Amazon Virtual Private Cloud (Amazon VPC). Ils sont accessibles directement par les instances Amazon EC2 ou d'autresAWS services déployés dans le même Amazon VPC. En outre, Amazon DocumentDB est accessible par des instances Amazon EC2 ou d'autresAWS services dans différents VPC de la mêmeAWS région ou dans d'autres régions via le peering de VPC.

Supposons toutefois que votre cas d'utilisation nécessite que vous (ou votre application) accédiez à vos ressources Amazon DocumentDB depuis l'extérieur du VPC du cluster. Ce sera le cas pour la plupart des utilisateurs qui n'exécutent pas leur application sur une machine virtuelle dans le même VPC que le cluster Amazon DocumentDB. Lorsque vous vous connectez depuis l'extérieur du VPC, vous pouvez utiliser le tunneling SSH (également appelé redirection de port) pour accéder à vos ressources Amazon DocumentDB.

Pour créer un tunnel SSH, vous devez disposer d'une instance Amazon EC2 s'exécutant dans le même Amazon VPC que le Amazon DocumentDB Documentmentmentmentmentmentmentmentmentmentmentmentmentmentmentmentmentmen Vous pouvez soit utiliser une instance EC2 existante dans le même VPC que votre cluster, soit en créer une. Vous pouvez configurer un tunnel SSH vers le cluster Amazon DocumentDB`sample-cluster.node.us-east-1.docdb.amazonaws.com` en exécutant la commande suivante sur votre ordinateur local :

```
ssh -i "ec2Access.pem" -L 27017:sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017 ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com -
N
```

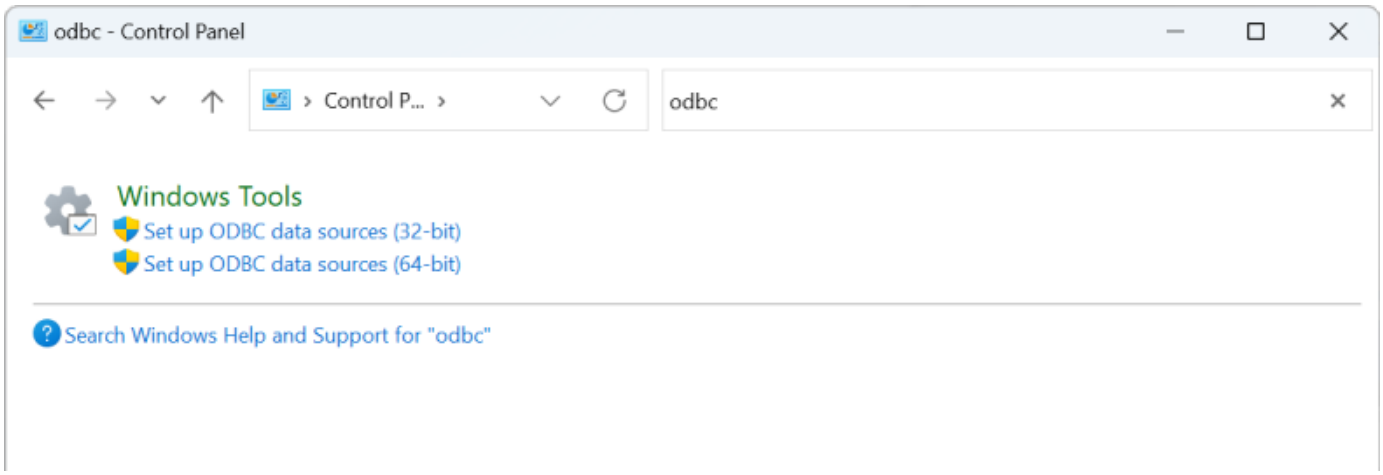
L'indicateur `-L` est utilisé pour réacheminer un port local. Il s'agit d'une condition préalable à la connexion à tout outil de BI exécuté sur un client extérieur à votre VPC. Une fois que vous avez exécuté l'étape ci-dessus, vous pouvez passer aux étapes suivantes pour l'outil de BI de votre choix.

Pour plus d'informations sur le tunneling SSH, consultez la documentation sur l'utilisation d'un tunnel SSH pour Connect à Amazon DocumentDB.

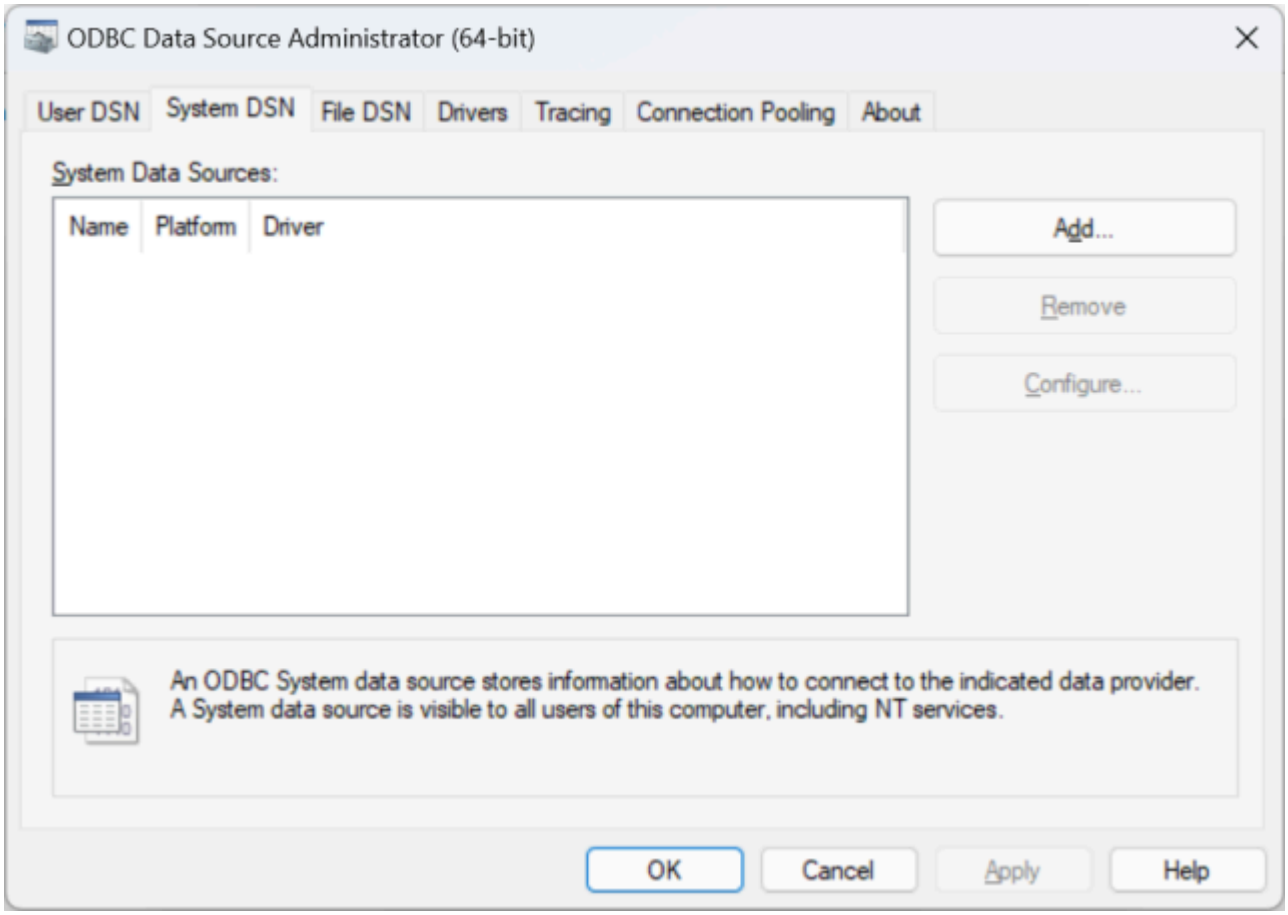
Configuration du pilote ODBC Amazon DocumentDB sous Windows

Suivez la procédure suivante pour configurer le pilote ODBC Amazon DocumentDB dans Windows :

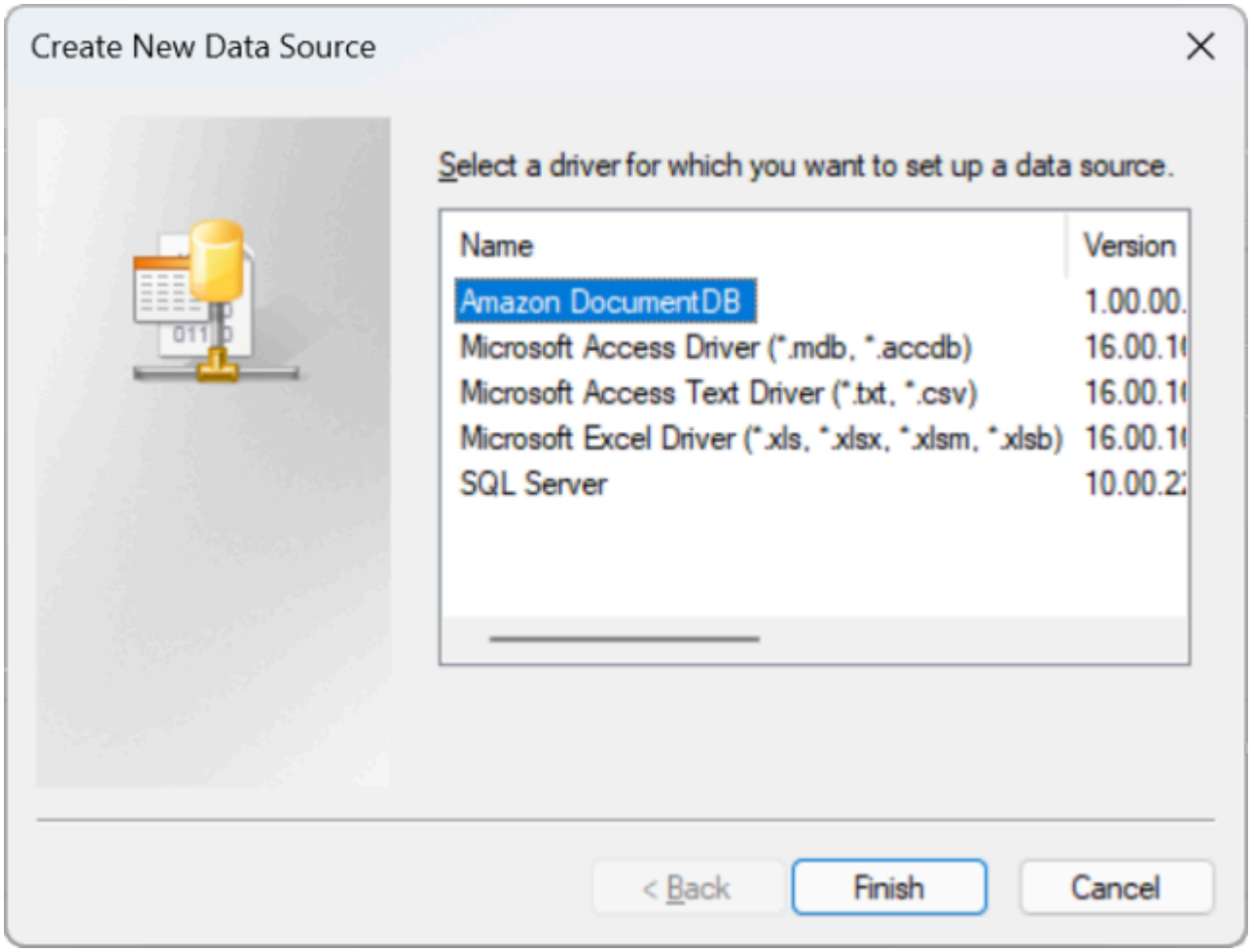
1. Ouvrez le Panneau de configuration dans Windows et recherchez ODBC (ou dans le menu, sélectionnez Outils Windows > Sources de données ODBC (32 bits) ou Sources de données ODBC (64 bits)) :



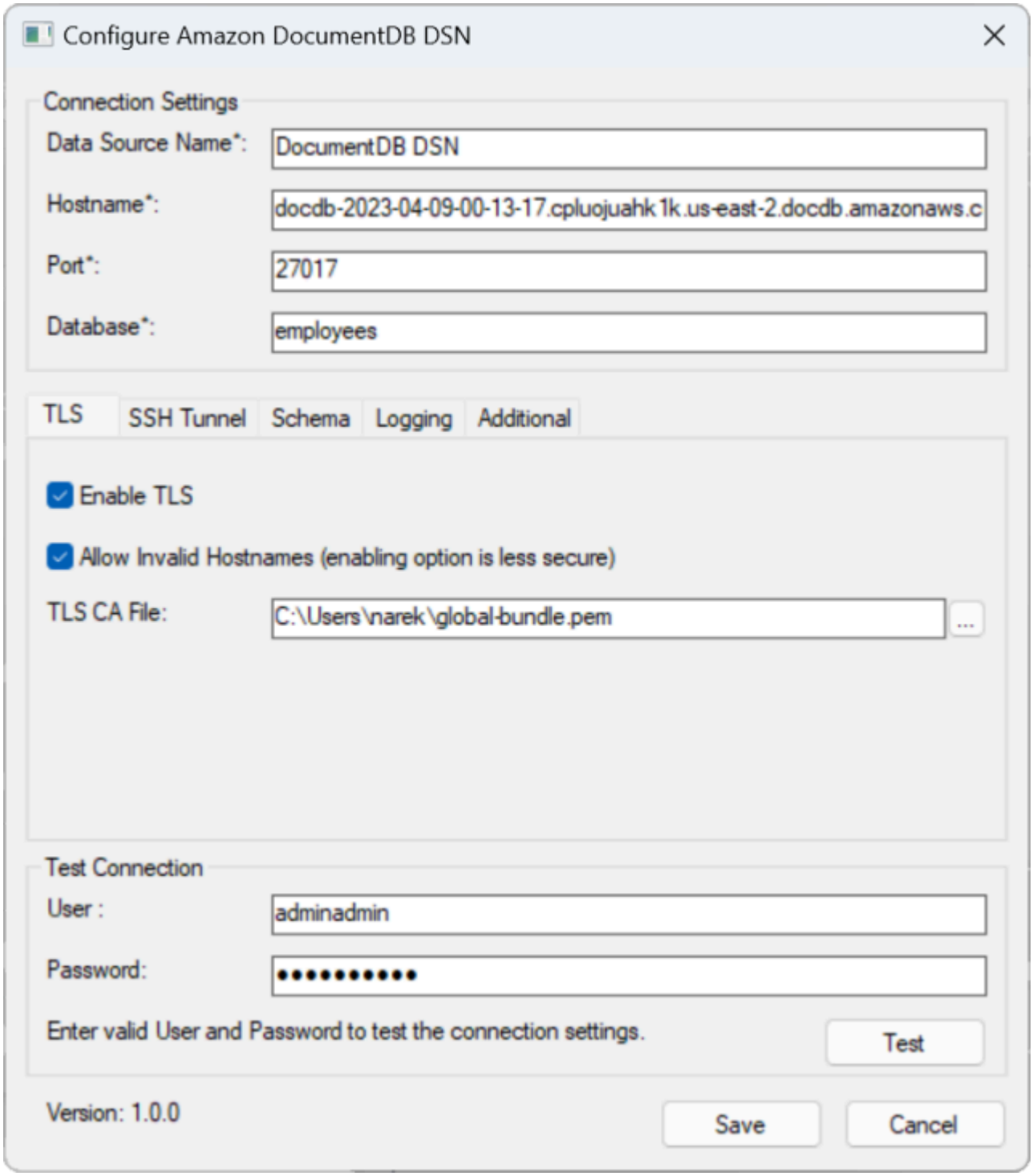
2. Sélectionnez l'administrateur de source de données du pilote ODBC approprié : optez pour la version 32 bits si elle est installée, sinon, choisissez la version 64 bits.
3. Sélectionnez l'onglet Sytem DSN, puis cliquez sur Ajouter... pour ajouter un nouveau DSN :



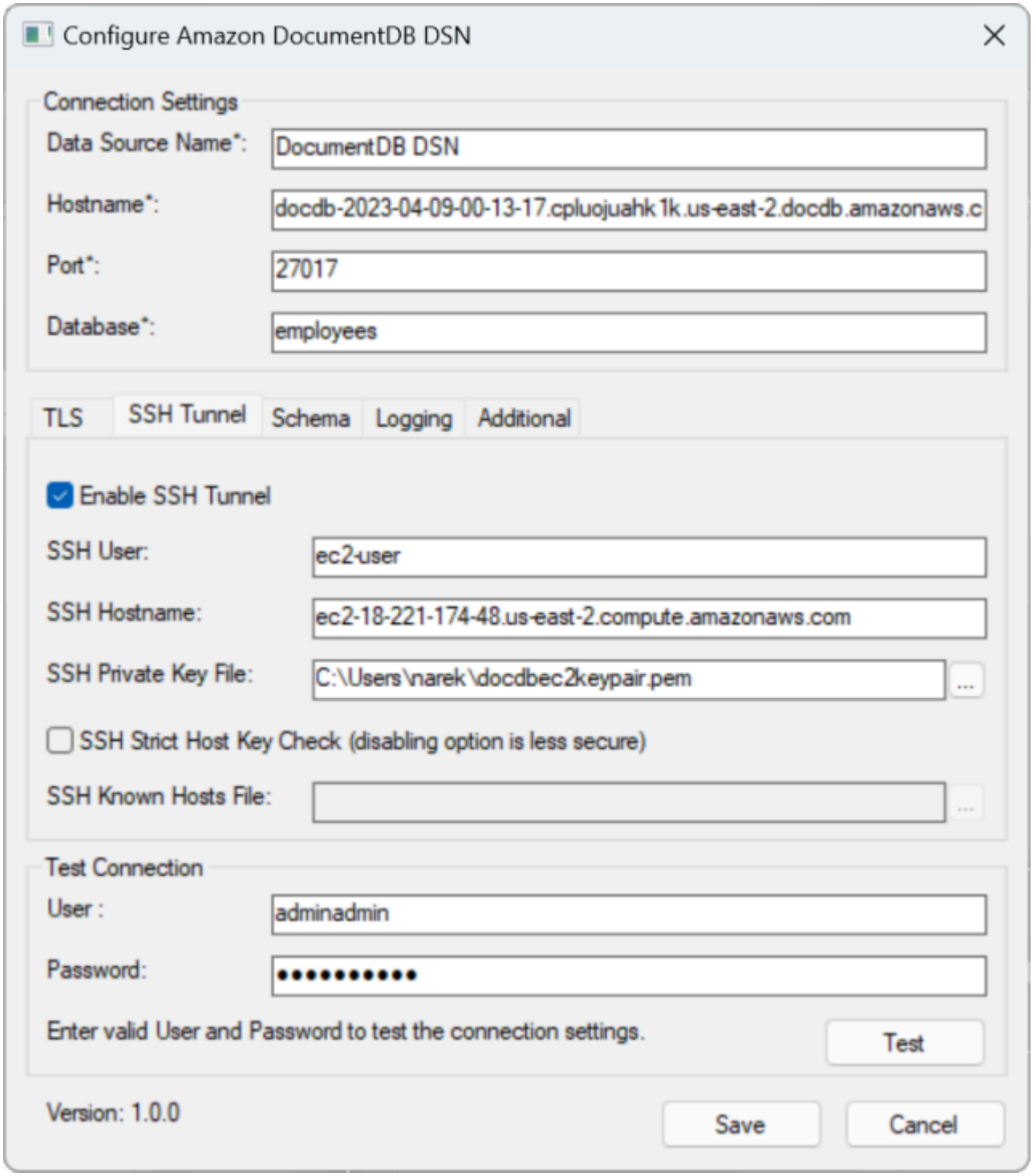
4. Choisissez Amazon DocumentDB dans la liste des pilotes de source de données :



5. Dans la boîte de dialogue Configurer Amazon DocumentDB DSN, renseignez les champs Paramètres de configuration, onglet TLS et Test de connexion, puis cliquez sur Enregistrer :



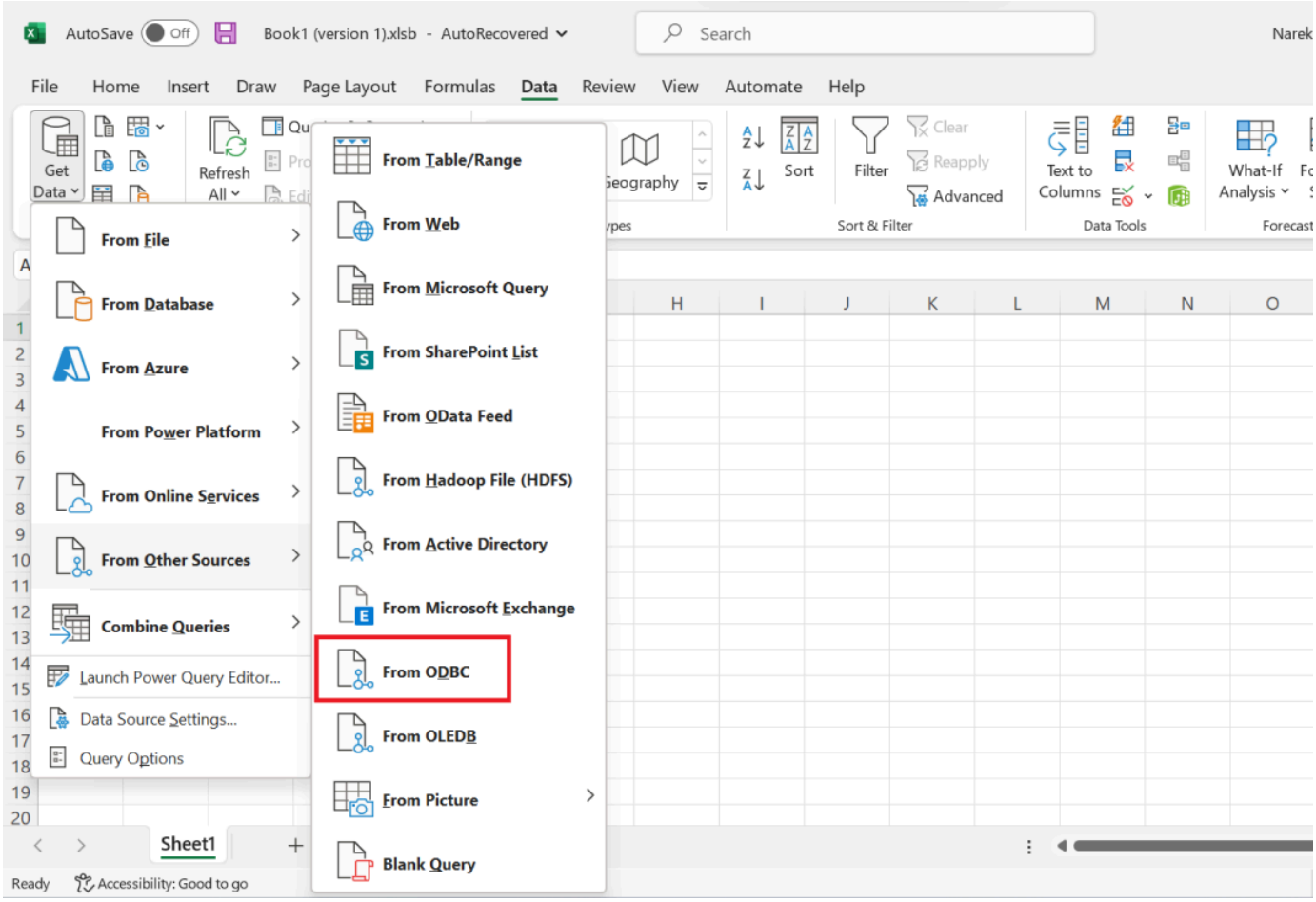
6. Assurez-vous de remplir correctement le formulaire Windows, car les détails de connexion varient en fonction de la méthode de tunneling SSH que vous avez choisie vers l'instance EC2. Consultez les méthodes de tunneling SSH ici. Reportez-vous à la section Syntaxe et options de la chaîne de connexion pour plus d'informations sur chaque propriété.



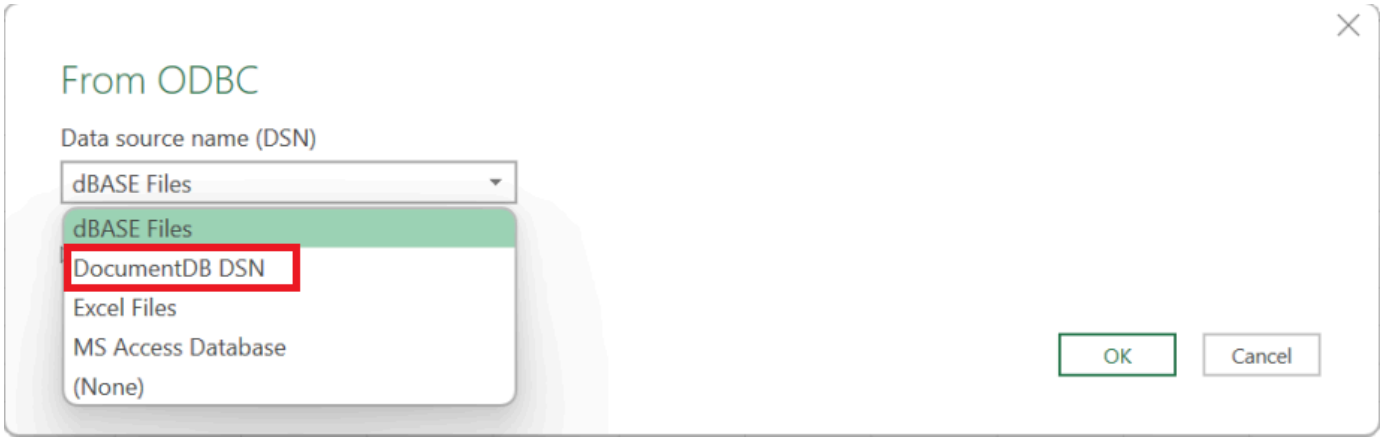
Pour plus d'informations sur la configuration du pilote ODBC Amazon DocumentDB sous Windows, cliquez ici.

Connect à Amazon DocumentDB depuis Microsoft Excel

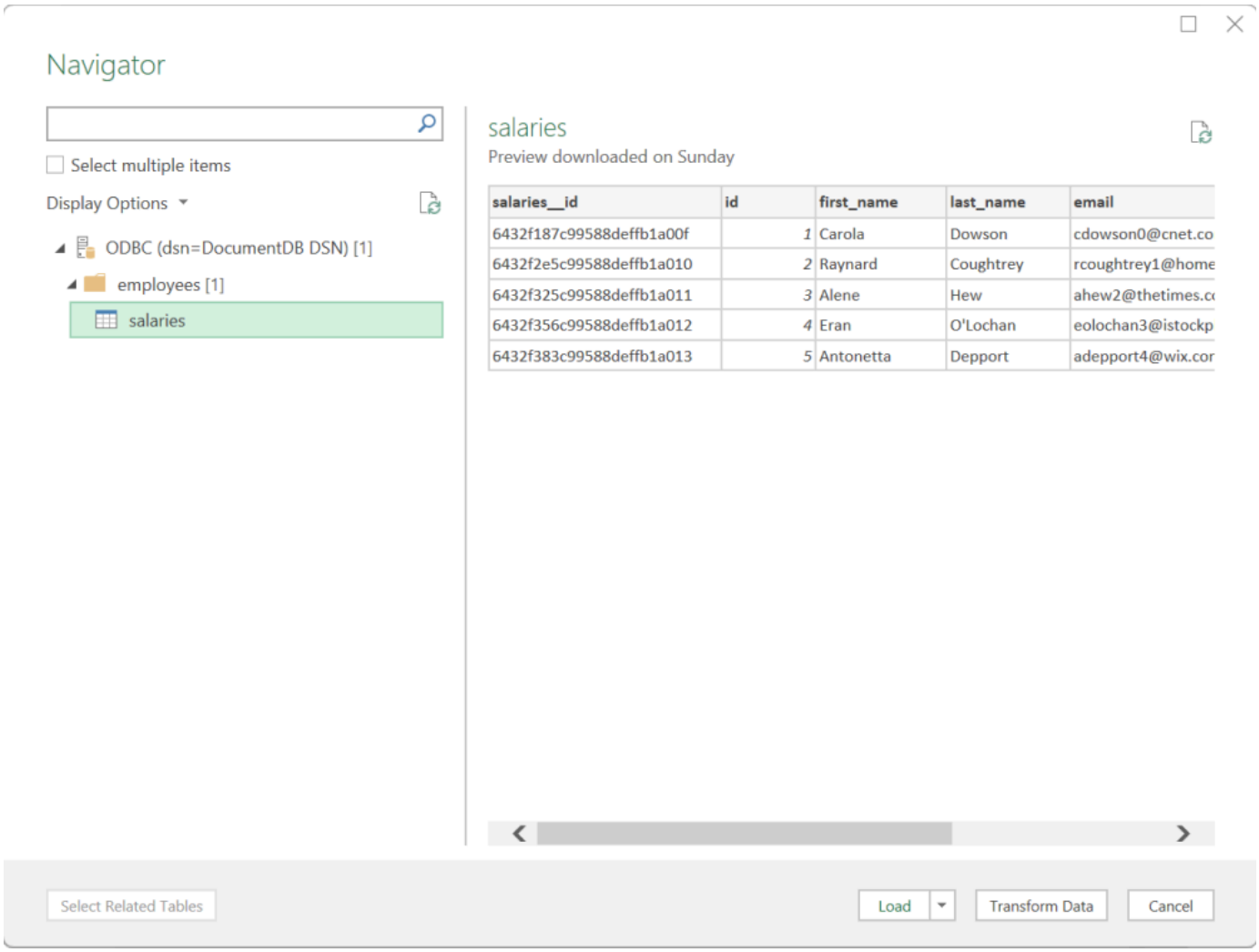
1. Assurez-vous que le pilote Amazon DocumentDB a été correctement installé et configuré. Pour plus d'informations, reportez-vous à la section Configuration du pilote ODBC sous Windows.
2. Lancez Microsoft Excel.
3. Accédez à Données > Obtenir des données > À partir d'autres sources.
4. Choisissez parmi ODBC :



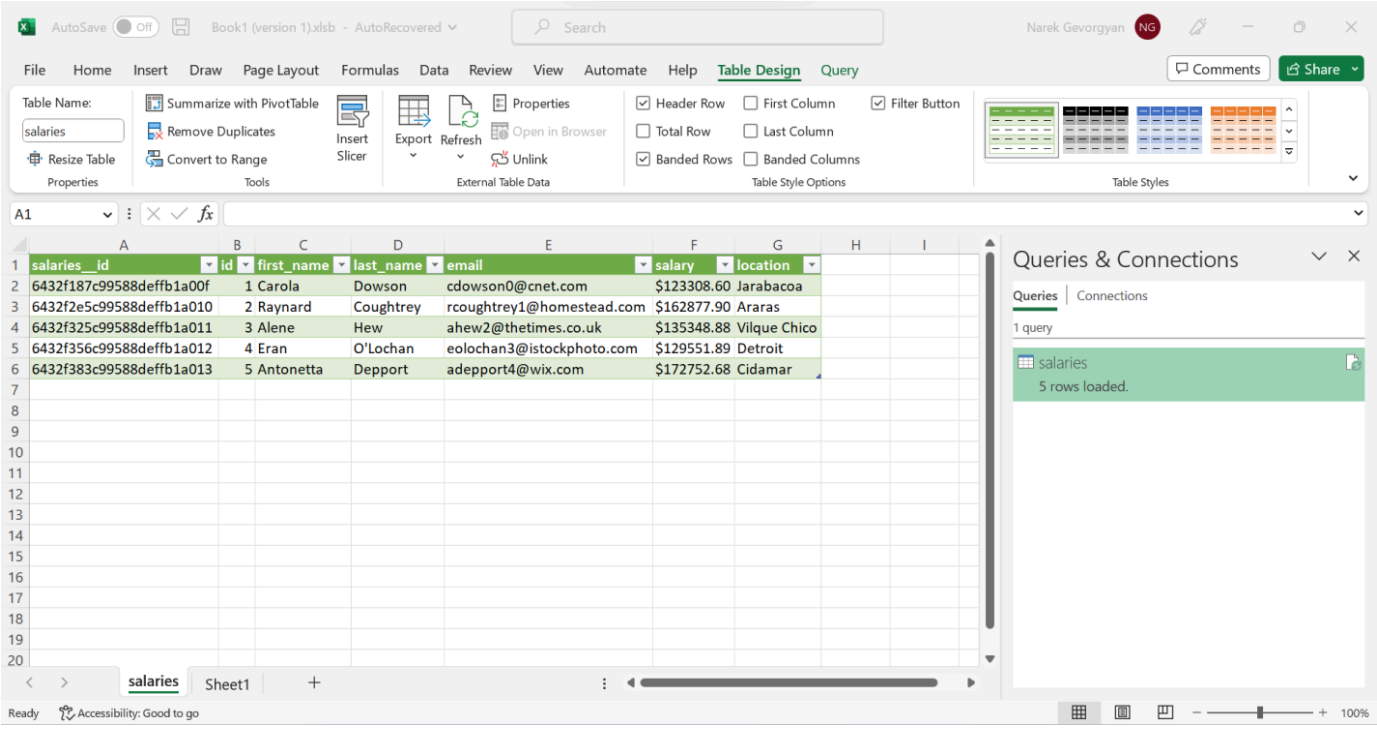
5. Sélectionnez la source de données dans le menu déroulant Nom de la source de données (DSN) associé à Amazon DocumentDB :



6. Sélectionnez la collection à partir de laquelle vous souhaitez charger des données dans Excel :



7. Charger des données dans Excel :



Connect à Amazon DocumentDB depuis Microsoft Power BI Desktop

Rubriques

Prérequis

Avant de commencer, assurez-vous que le pilote ODBC Amazon DocumentDB est correctement installé.

Ajout d'un connecteur personnalisé Microsoft Power BI Desktop

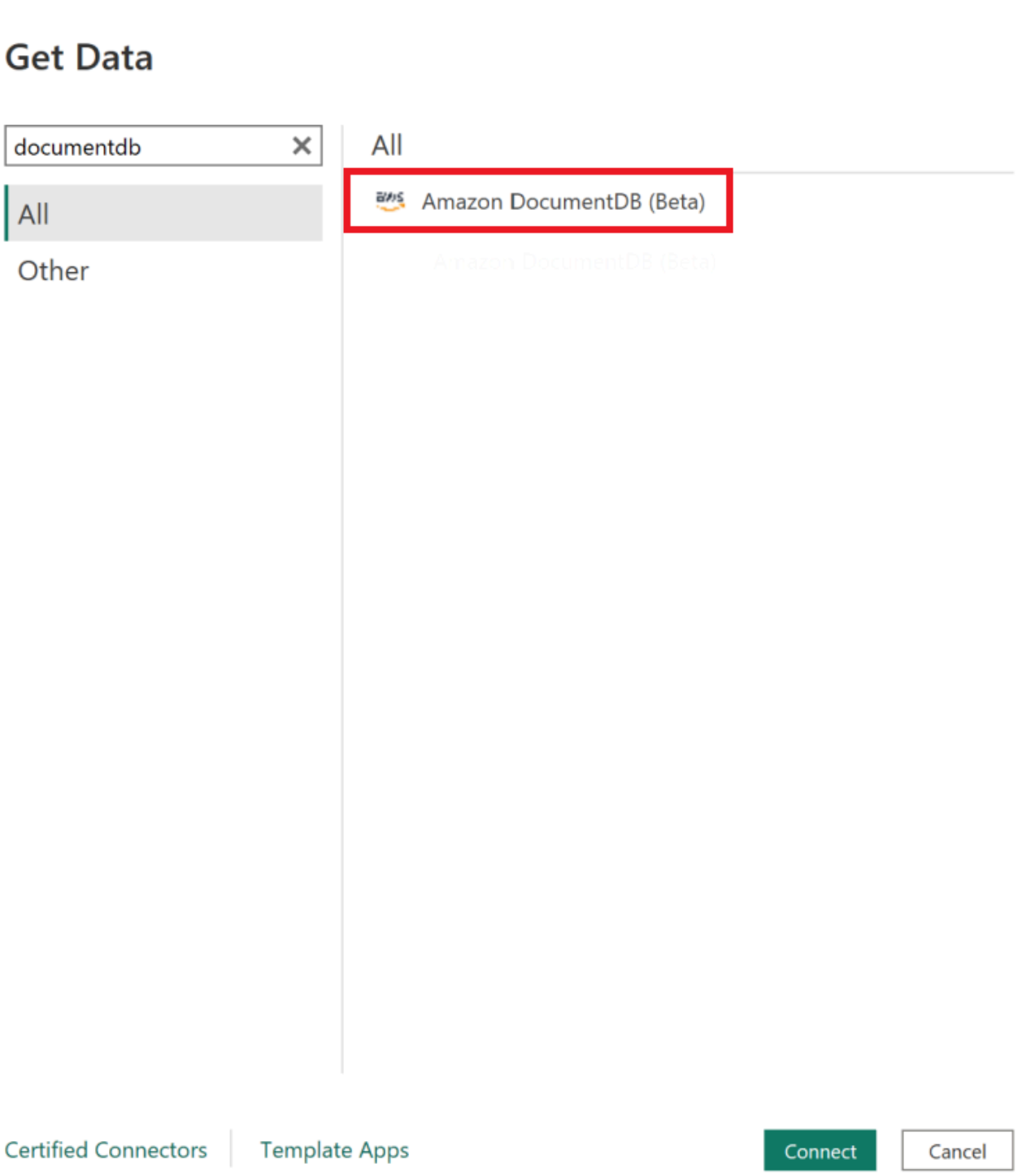
Copiez le`AmazonDocumentDBConnector.mez` fichier`<User>\Documents\Power BI Desktop \Custom Connectors\` dans le dossier (ou`<User>\OneDrive\Documents\Power BI Desktop\Custom Connectors` si vous l'utilisez OneDrive). Cela permettra à Power BI d'accéder

à un connecteur personnalisé. Vous pouvez obtenir le connecteur vers Power BI Desktop ici. Redémarrez Power BI Desktop pour vous assurer que le connecteur est chargé.

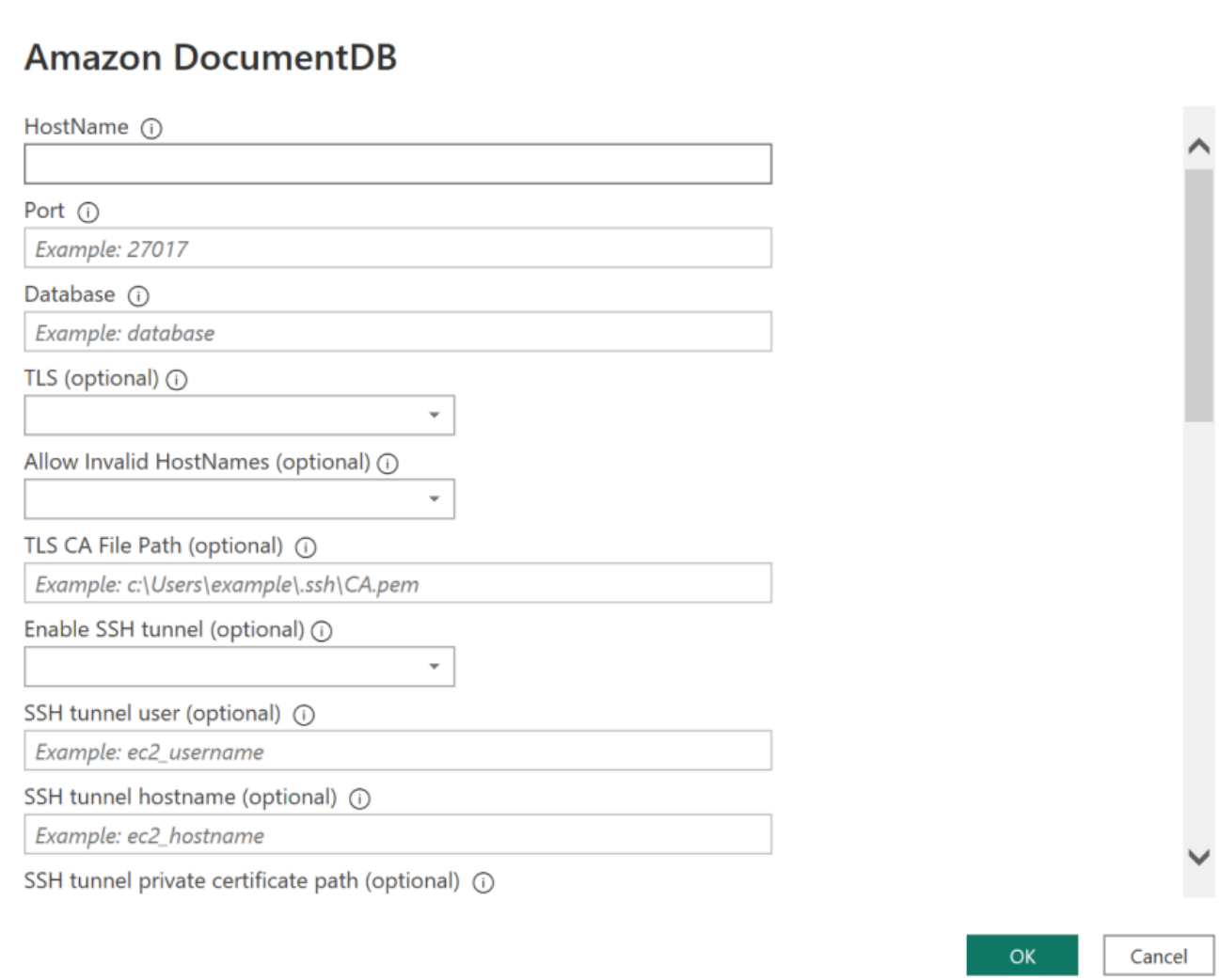
> **Note**
>
> Le connecteur personnalisé prend uniquement en charge le nom d'utilisateur et le mot de passe Amazon DocumentDB à des fins d'authentification.

Connexion à l'aide du connecteur personnalisé Amazon DocumentDB

1. Sélectionnez Amazon DocumentDB (bêta) dans Obtenir des données et cliquez sur Connect. Si vous recevez un avertissement concernant l'utilisation d'un service tiers, cliquez sur Continuer.



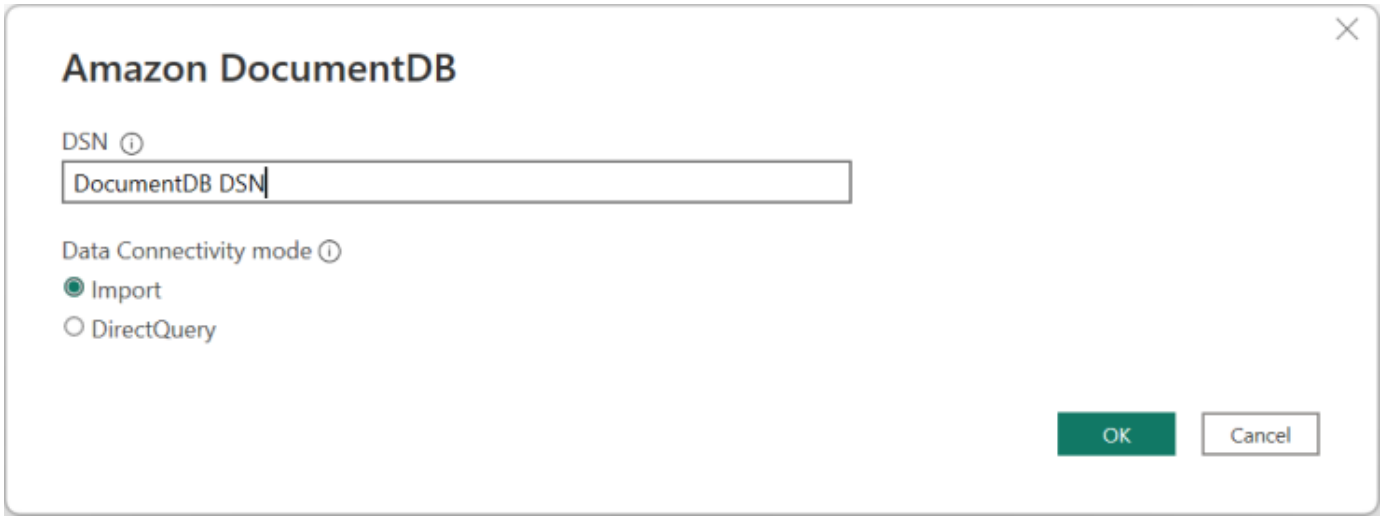
2. Entrez toutes les informations nécessaires pour vous connecter à votre cluster Amazon DocumentDB, puis cliquez sur OK :



> **Note**
>
> Selon la configuration du nom de source de données (DSN) de votre pilote ODBC, l'écran des détails de la connexion SSH peut ne pas s'afficher si vous avez déjà fourni les informations nécessaires dans les paramètres du DSN.

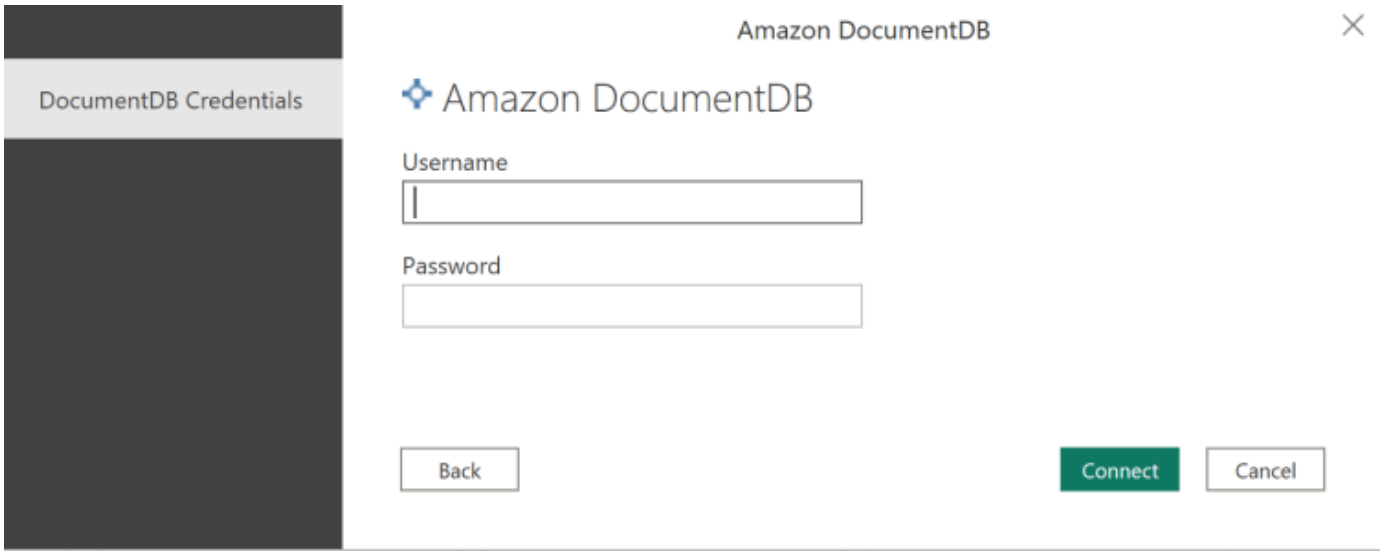
3. Choisissez le mode de connectivité des données :

- Importer : charge toutes les données et stocke les informations sur le disque. Les données doivent être actualisées et rechargées afin d'afficher les mises à jour des données.
- Requête directe : ne charge pas les données, mais effectue des requêtes en direct sur les données. Cela signifie que les données n'ont pas besoin d'être actualisées et rechargées pour afficher les mises à jour des données.

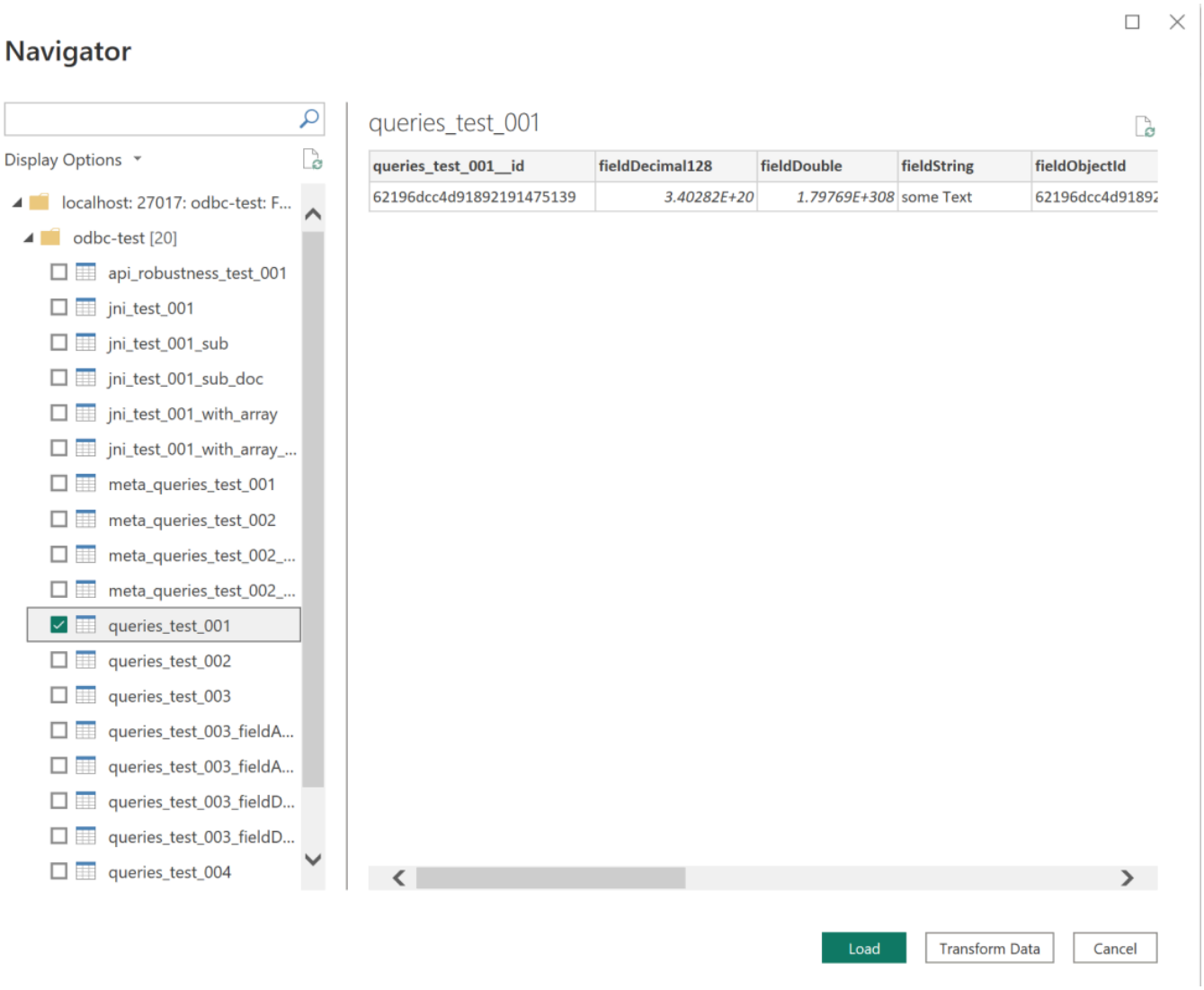


> **Note**
>
> Si vous utilisez un jeu de données très volumineux, l'importation de toutes les données peut prendre plus de temps.

4. S'il s'agit de la première connexion à cette source de données, sélectionnez le type d'authentification et saisissez vos informations d'identification lorsque vous y êtes invité. Cliquez ensuite sur Connect :



5. Dans la boîte de dialogue du Navigateur, sélectionnez les tables de base de données souhaitées, puis cliquez sur Charger pour charger les données ou sur Transformer les données pour poursuivre la transformation des données.



> **Note**
>
> Les paramètres de votre source de données sont enregistrés une fois que vous vous connectez. Pour les modifier, sélectionnez Transformer les données > Paramètres de la source de données.

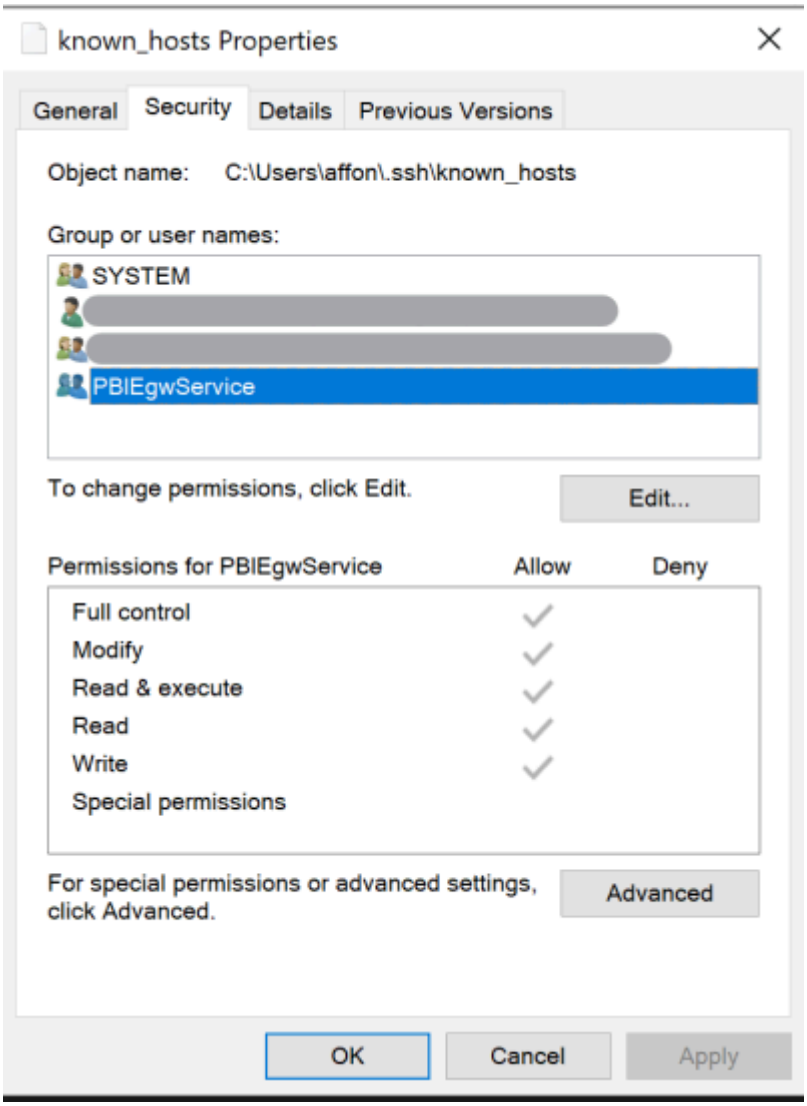
Configuration de Microsoft Power BI Gateway

Prérequis :

- Assurez-vous que le connecteur personnalisé fonctionne avec Power BI Gateway.

- Assurez-vous que le DSN ODBC est créé dans les sources de données ODBC de l'onglet Système de la machine sur laquelle Power BI Gateway est installé.

Si vous utilisez la fonctionnalité de tunnel SSH interne, le fichier`known_hosts` doit se trouver à un endroit où le compte de service Power BI y a accès.



> **Note**
>
> Cela s'applique également à tous les fichiers dont vous pourriez avoir besoin pour établir une connexion à votre cluster Amazon DocumentDB, tels qu'un fichier de certificat d'autorité de certification (CA) (fichier pem).

Génération automatique de schémas

Le pilote ODBC utilise le pilote JDBC Amazon DocumentDB via JNI (Java Native Interface), ce qui permet à la fonctionnalité de génération automatique de schémas de fonctionner de la même manière dans le pilote JDBC. Pour plus d'informations sur la génération automatique de schémas, consultez la section Génération automatique de schémas JDBC. De plus, pour en savoir plus sur l'architecture du pilote ODBC, cliquez ici.

Support SQL et limitations

Le pilote ODBC Amazon DocumentDB est un pilote en lecture seule qui prend en charge un sous-ensemble de SQL-92 et certaines extensions courantes. Reportez-vous à la documentation relative au support et aux limites ODBC pour plus d'informations.

Résolution des problèmes

Si vous rencontrez des problèmes lors de l'utilisation du pilote ODBC Amazon DocumentDB, consultez le Guide de résolution des problèmes.

Quotas et limites Amazon DocumentDB

Cette rubrique décrit les quotas de ressources, les limites et les contraintes de dénomination pour Amazon DocumentDB (avec compatibilité avec MongoDB).

Pour certaines fonctionnalités de gestion, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon Relational Database Service (Amazon RDS) et Amazon Neptune.

Rubriques

Types d'instance pris en charge

Amazon DocumentDB prend en charge les instances à la demande et les types d'instances suivants :

- Mémoire optimisée :
  - Types d'instances R6G :`db.r6g.large`,,`db.r6g.2xlarge`,, `db.r6g.4xlargedb.r6g.8xlarge`,`db.r6g.12xlarge`. `db.r6g.16xlarge`
  - Types d'instances R5 :`db.r5.large`,`db.r5.2xlarge`,,`db.r5.4xlarge`, `db.r5.8xlargedb.r5.12xlarge`, `db.r5.16xlargedb.r5.24xlarge`.
  - Types d'instance R4 : `db.r4.large`, `db.r4.2xlarge`, `db.r4.4xlarge`, `db.r4.8xlarge`, `db.r4.16xlarge`.

- Performances éclatantes :
  - Types d'instances T4G : `db.t4g.medium`.
  - Types d'instances T3 : `db.t3.medium`.

Pour de plus amples informations sur les types d'instance pris en charge et leurs spécifications, veuillez consulter Spécifications de la classe d'instance.

Régions prises en charge

Amazon DocumentDB est disponible dans les régions suivantes : AWS

Nom de la région	Région	Zones de disponibilité (calcul)
USA Est (Ohio)	`us-east-2`	3
USA Est (Virginie du Nord)	`us-east-1`	6
USA Ouest (Oregon)	`us-west-2`	4
Amérique du Sud (São Paulo)	`sa-east-1`	3
Asie-Pacifique (Hong Kong)	`ap-east-1`	3
Asie-Pacifique (Hyderabad)	`ap-south-2`	3
Asie-Pacifique (Mumbai)	`ap-south-1`	3
Asie-Pacifique (Séoul)	`ap-northeast-2`	4
Asie-Pacifique (Singapour)	`ap-southeast-1`	3
Asie-Pacifique (Sydney)	`ap-southeast-2`	3
Asie-Pacifique (Tokyo)	`ap-northeast-1`	3
Canada (Centre)	`ca-central-1`	3
Région Chine (Beijing)	`cn-north-1`	3

Nom de la région	Région	Zones de disponibilité (calcul)
Chine (Ningxia)	`cn-northwest-1`	3
Europe (Francfort)	`eu-central-1`	3
Europe (Irlande)	`eu-west-1`	3
Europe (Londres)	`eu-west-2`	3
Europe (Milan)	`eu-south-1`	3
Europe (Paris)	`eu-west-3`	3
Moyen-Orient (EAU)	`me-central-1`	3
AWS GovCloud (US-Ouest)	`us-gov-west-1`	3
AWS GovCloud (USA Est)	`us-gov-east-1`	3

Quotas régionaux

Pour certaines fonctionnalités de gestion, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon Relational Database Service (Amazon RDS). Le tableau suivant contient les limites régionales partagées entre Amazon DocumentDB et Amazon RDS.

> **Note**
>
> La technologie partagée Amazon RDS décrite ci-dessus s'applique uniquement aux clusters basés sur des instances Amazon DocumentDB. Les clusters élastiques Amazon DocumentDB ne partagent pas de technologie avec Amazon RDS.

Les limites suivantes s'appliquent aux clusters basés sur des instances Amazon DocumentDB et s'appliquent par compte et par région. AWS

Ressource	AWS limite par défaut
Clusters	40
Groupes de paramètres de cluster	50
Abonnements aux événements	20
Instances	40
Instantanés de cluster manuels	100
Réplicas en lecture par cluster	15
Groupes de sous-réseaux	50
Sous-réseaux par groupe de sous-réseaux	20
Balises par ressource	50
Groupes de sécurité VPC par instance	5

Les limites suivantes s'appliquent aux clusters élastiques Amazon DocumentDB et s'appliquent par AWS compte et par région.

Ressource	AWS limite par défaut
Clusters élastiques	20
Clusters élastiques (vCPU)	1 024
Instantané manuel du cluster élastique	20

Vous pouvez utiliser Service Quotas pour demander l'augmentation d'un quota, si ce dernier est ajustable. Certaines demandes sont automatiquement résolues, tandis que d'autres sont soumises à AWS Support. Vous pouvez suivre le statut d'une demande d'augmentation de quota soumise à AWS Support. Les demandes d'augmentation des quotas de service ne reçoivent pas de soutien prioritaire.

Si vous avez une demande urgente, veuillez contacter AWS Support. Pour plus d'informations sur les quotas de service, consultez Qu'est-ce que les quotas de service ?

Pour demander une augmentation du quota pour Amazon DocumentDB :

1. Ouvrez la console Service Quotas sur https://console.aws.amazon.com/servicequotas et, si nécessaire, connectez-vous.
2. Dans le panneau de navigation, choisissez Services AWS .
3. Sélectionnez Amazon DocumentDB (compatible avec MongoDB) ou Amazon DocumentDB Elastic Cluster dans la liste, ou saisissez l'un ou l'autre dans le champ de recherche.
4. Si le quota est ajustable, vous pouvez sélectionner son bouton radio ou son nom, puis choisir Request quota increase (Demander une augmentation du quota) dans le coin supérieur droit de la page.
5. Pour Change quota value (Modifier la valeur du quota), saisissez la nouvelle valeur. Elle doit être supérieure à la valeur actuelle.
6. Choisissez Request (Demander). Une fois la demande résolue, la Applied quota value (Valeur de quota appliquée) pour le quota est définie selon la nouvelle valeur.
7. Pour afficher les demandes en attente ou récemment résolues, choisissez Dashboard (Tableau de bord) dans le volet de navigation. Pour les demandes en attente, choisissez l’état de la demande pour ouvrir le reçu de la demande. L’état initial d'une demande est `Pending`. Une fois le statut changé en`Quota requested`, vous verrez le numéro de dossier avec AWS Support. Choisissez le numéro de dossier pour ouvrir le billet pour votre demande.

Restriction de regroupement

Le tableau suivant décrit les limites d'agrégation dans Amazon DocumentDB.

Ressource	Limite
Nombre maximum d'étapes prises en charge	500

Limites du cluster

Le tableau suivant décrit les limites de cluster basées sur les instances Amazon DocumentDB.

Ressource	Limite
Taille du cluster (somme de tous les collections et index)	128 Tio
Taille de la collection (la somme de toutes les collections ne peut pas être supérieure à la limite de cluster) - n'inclut pas la taille de l'index	32 To
Collections par cluster	100 000
Bases de données par cluster	100 000
Taille de la base de données (la somme de toutes les bases de données ne peut pas être supérieure à la limite de cluster)	128 Tio
Profondeur d'imbrication des documents	200 niveaux
Taille des documents	16 Mo
Taille de clé d'index	2 048 octets
Index par collection	64
Clés dans un index composé	32
Nombre maximal d'écritures dans une seule commande par lots	100 000

Ressource	Limite
Nombre d'utilisateurs par cluster	1 000

Limites d’instance

Le tableau suivant décrit les limites Amazon DocumentDB par instance.

Type d'instance	Mémoire d'instance (GiB)	Connexions (toutes)	Limite du curseur	Transactions ouvertes	Connexions (actives)
T3. Medium	4	500	30	50	102
T4G. Moyen	4	500	30	50	102
R4. Grand	15,25	1700	450	N/A	1100
R4.xLarge	30,5	3400	450	N/A	2700
R4.2 x grand	61	6800	450	N/A	4500
R4.4 x grand	122	13600	725	N/A	4500
R4,8 x grand	288	27200	1450	N/A	4500
R4,16 x large	488	30 000	2900	N/A	4500
R5. Grand	16	1700	450	200	1100
R5.xLarge	32	3500	450	400	2700
R5,2 x grand	64	7100	450	800	4500
R5,4 x grand	128	14200	760	1600	4500
R5,8 x grand	256	28400	1520	3200	4500
R5,12 x large	383	30 000	2280	4800	4500
R5.16 x large	512	30 000	3040	6400	4500
R 5,24 x large	768	30 000	4560	9600	4500

Type d'instance	Mémoire d'instance (GiB)	Connexions (toutes)	Limite du curseur	Transactions ouvertes	Connexions (actives)
R6G. Grand	16	1700	450	200	1100
R6G.xLarge	32	3500	450	400	2700
R6G, 2 x grand	64	7100	450	800	4500
R6G.4XLarge	128	14200	760	1600	4500
R6 G, 8 x L	256	28400	1520	3200	4500
R6 G. 12 x large	383	30 000	2280	4800	4500
R6 G. 16 x large	512	30 000	3040	6400	4500

Vous pouvez surveiller et déclencher une alarme sur les limites par instance à l'aide CloudWatch des métriques suivantes. Pour en savoir plus sur les CloudWatch métriques Amazon DocumentDB, consultez. Surveillance d'Amazon DocumentDB avec CloudWatch

Limite	CloudWatch Métriques
Mémoire d'instance	FreeableMemory
Connexions	DatabaseConnectionsMax
Curseurs	DatabaseCursorsMax
Transactions	TransactionsOpenMax

Contraintes d'affectation de noms

Le tableau suivant décrit les contraintes de dénomination dans Amazon DocumentDB.

Ressource	Limite par défaut
Identifiant du cluster	• La longueur est de [1 à 63] lettres, chiffres ou traits d'union. • Le premier caractère doit être une lettre. • Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs. • Doit être unique pour tous les clusters (sur Amazon RDS, Amazon Neptune et Amazon DocumentDB) AWS par compte et par région.
Nom de la collection : `<col>`	La longueur est de [1 à 57] caractères.
Nom de la base de données : `<db>`	La longueur est de [1 à 63] caractères.
Nom complet de la collection :`<db>.<col>`	La longueur est de [3 à 120] caractères.
Nom complet de l'index : `<db>.<col>.$<index>`	La longueur est de [6 à 127] caractères.
Nom de l'index : `<col>$<index>`	La longueur est de [3 à 63] caractères.
Identifiant de l'instance	• La longueur est de [1 à 63] lettres, chiffres ou traits d'union • Le premier caractère doit être une lettre

Ressource	Limite par défaut
	• Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs • Doit être unique pour toutes les instances (sur Amazon RDS, Amazon Neptune et Amazon DocumentDB) AWS par compte et par région.

Mot de passe principal	• Entre 8 et 100 caractères ASCII imprimables.
	• Tous les caractères ASCII imprimables peuvent être utilisés, à l'exception des suivants :
	  • **/** (barre oblique)
	  • **"** (guillemets doubles)
	  • **@** (symbole arobase)

Nom d'utilisateur principal	• Entre 1 et 63 caractères alphanumériques. • Le premier caractère doit être une lettre. • Ne peut pas être un mot réservé du moteur de base de données.
Nom du groupe de paramètres	• Entre 1 et 255 caractères alphanumériques. • Le premier caractère doit être une lettre. • Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

Contraintes de durée de vie (TTL)

Les suppressions depuis un index TTL ne sont pas garanties au cours d'une période spécifique et sont effectuées dans la mesure du possible. Des facteurs tels que l'utilisation de ressources d'instance, la taille de document et le débit global peuvent affecter le moment d'une suppression TTL.

Limites de cluster élastiques

Le tableau suivant décrit les limites maximales dans les clusters élastiques Amazon DocumentDB.

Ressource	Limite
Clusters élastiques par région	20
vCPU additionné sur tous les clusters élastiques par région	1 024
Instantanés manuels du cluster par région	20
Partitions par cluster	32
Stockage par cluster (lorsque les données sont réparties uniformément par clé de partition)	4 PiB
Connexions au cluster	La valeur inférieure de 300 000 ou le nombre de partitions x la limite de connexion associée au vCPU par partition
UnSharded taille de la collection	32 TO
Taille de collection fractionnée (lorsque les données sont réparties uniformément par clé de partition)	1 PB
Bases de données par cluster	10 000
UnSharded collections par cluster	100 000
Collections partitionnées par cluster	1 000

Ressource	Limite
Utilisateurs par cluster	100
Écrit en une seule commande par lots	100 000
Index par collection	64
Profondeur d'imbrication des documents	100 niveaux
Taille des documents	16 Mo
Taille de clé d'index	2048 bytes
Clés dans un index composé	32

Limites de partage des clusters élastiques

Le tableau suivant décrit les limites maximales de partitions dans les clusters élastiques Amazon DocumentDB.

Ressource	Limite
vCPU par instance de partition	64
Instances par partition	16
Stockage par partition	128 Tio
Stockage par collection par partition	32 TO

Limites de processeur, de mémoire, de connexion et de curseur du cluster élastique par partition

Le tableau suivant décrit les limites maximales de processeur, de mémoire, de connexion et de curseur dans les partitions de clusters élastiques Amazon DocumentDB.

vCPU par partition	Mémoire d'instance (GiB)	Limite de connexion	Limite du curseur
2	16	1700	450
4	32	3500	450
8	64	7100	450
16	128	14200	760
32	256	28400	1520
48	383	30 000	2280
64	512	30 000	3040

Interrogation

Cette section explique tous les aspects des requêtes avec Amazon DocumentDB.

Rubriques

Interrogation de documents

Il arrive que vous ayez besoin de consulter le stock de votre boutique en ligne, pour que les clients puissent voir et acheter ce que vous vendez. Interroger une collection est relativement simple, qu'il s'agisse de tous les documents dans la collection ou uniquement ceux qui répondent à un critère particulier.

Pour interroger des documents, utilisez l'opération `find()`. La commande `find()` possède un seul paramètre de document qui définit les critères à utiliser pour choisir le documents à renvoyer Le résultat obtenu à partir de `find()` est d'un document formaté en une seule ligne de texte sans sauts de ligne. Pour formater le document de sortie afin d'en faciliter la lecture, utilisez `find().pretty()`. Tous les exemples de cette rubrique utilisent `.pretty()` pour mettre en forme les données de sortie.

Les exemples de code suivants utilisent les quatre documents que vous avez insérés dans la `example` collection au cours des deux exercices précédents `insertOne()` et `insertMany()` qui se trouvent dans la section Ajouter des documents de la section Travailler avec des documents.

Rubriques

- Récupération de documents correspondant à un tableau
- Extraction de documents correspondant à une valeur d'un tableau
- Récupération de documents à l'aide d'opérateurs

Récupération de tous les documents d'une collection

Pour récupérer tous les documents de votre collection, utilisez l'opération `find()` avec un document de requête vide.

La requête suivante renvoie tous les documents de la collection `example`.

```
db.example.find( {} ).pretty()
```

Récupération de documents correspondant à une valeur de champ

Pour récupérer tous les documents qui correspondent à un champ et à une valeur, utilisez l'opération `find()` avec un document de requête qui identifie les champs et les valeurs à faire correspondre.

En utilisant les documents précédents, cette requête renvoie tous les documents où le champ « Item (Élément) » est défini sur « Pen ».

```
db.example.find( { "Item": "Pen" } ).pretty()
```

Récupération de documents correspondant à un document intégré

Pour rechercher tous les documents qui correspondent à un document intégré, utilisez l'opération `find()` avec un document de requête qui spécifie le nom du document intégré et tous les champs et les valeurs de ce document intégré.

Lorsqu'une correspondance est établie avec un document intégré , le nom de ce document doit être identique à celui spécifié dans la requête. De plus, les champs et les valeurs dans le document intégré doivent correspondre à la requête.

La requête suivante renvoie uniquement le document « Poster Paint ». Cela est dû au fait que « Pen » a des valeurs différentes pour « `OnHand` » et « `MinOnHand` », et que et « Spray Paint » a un champ de plus (`OrderQnty`) que le document de requête.

```
db.example.find({"Inventory": {
    "OnHand": 47,
```

```
    "MinOnHand": 50 } } ).pretty()
```

Extraction de documents correspondant à une valeur de champ dans un document intégré

Pour rechercher tous les documents qui correspondent à un document intégré, utilisez l'opération `find()` avec un document de requête qui spécifie le nom du document intégré et tous les champs et les valeurs de ce document intégré.

Au vu des documents précédents, la requête suivante utilise la « notation de points » pour spécifier le document intégré et les champs d'intérêt. Tout document correspondant est renvoyé, quels que soient les autres champs présents dans le document intégré. La requête renvoie « Poster Paint » et « Spray Paint », car ils correspondent tous deux aux champs et aux valeurs spécifiés.

```
db.example.find({"Inventory.OnHand": 47, "Inventory.MinOnHand": 50 }).pretty()
```

Récupération de documents correspondant à un tableau

Pour rechercher tous les documents qui correspondent à un tableau, utilisez l'opération `find()` avec le nom du tableau qui vous intéresse et toutes les valeurs de ce tableau. La requête renvoie tous les documents ayant un tableau avec ce nom et dans lequel les valeurs sont identiques et figurent dans le même ordre que dans la requête.

La requête suivante renvoie uniquement « Pen », car « Poster Paint » a une couleur supplémentaire (blanc) et les couleurs de « Spray Paint » figurent dans un ordre différent.

```
db.example.find( { "Colors": ["Red","Green","Blue","Black"] } ).pretty()
```

Extraction de documents correspondant à une valeur d'un tableau

Pour rechercher tous les documents contenant une valeur de tableau particulière, utilisez l'opération `find()` avec le nom du tableau et la valeur qui vous intéressent.

```
db.example.find( { "Colors": "Red" } ).pretty()
```

L'opération précédente renvoie les trois documents, car chacun d'entre eux possède un tableau nommé `Colors` et la valeur « `Red` » dans le tableau. Si vous spécifiez la valeur « `White` », la requête renvoie uniquement « Poster Paint ».

Récupération de documents à l'aide d'opérateurs

La requête suivante renvoie tous les documents où la valeur « `Inventory.OnHand` » est inférieure à 50.

```
db.example.find(
        { "Inventory.OnHand": { $lt: 50 } } )
```

Pour obtenir une liste des opérateurs de requête pris en charge, consultez Opérateurs de projection et de requête.

Plan de requête

Comment puis-je voir **`executionStats`** pour un plan de requête ?

Lorsque vous déterminez pourquoi une requête s'exécute plus lentement que prévu, il peut être utile de comprendre ce que représente `executionStats` pour le plan de requête. `executionStats` fournit le nombre de documents renvoyés à partir d'une étape particulière (`nReturned`), le temps d'exécution passé à chaque étape (`executionTimeMillisEstimate`) et le temps nécessaire à la génération d'un plan de requête (`planningTimeMillis`). Vous pouvez déterminer les étapes les plus longues de votre requête pour vous aider à concentrer vos efforts d'optimisation à partir de la sortie de `executionStats`, comme illustré dans les exemples de requête ci-dessous. Le paramètre `executionStats` ne prend pas actuellement en charge les commandes `update` et `delete`.

> **Note**
>
> Amazon DocumentDB émule l'API MongoDB 3.6 sur un moteur de base de données spécialement conçu qui utilise un système de stockage distribué, tolérant aux pannes et autoréparateur. Par conséquent, les plans de requête et le résultat de `explain()` peuvent différer entre Amazon DocumentDB et MongoDB. Les clients qui souhaitent contrôler leur plan de requête peuvent utiliser l'opérateur `$hint` pour appliquer la sélection d'un index préféré.

Exécutez la requête que vous souhaitez améliorer sous la commande `explain()` comme suit.

```
db.runCommand({explain: {query document}}).
```

```
explain("executionStats").executionStats;
```

Voici un exemple d'opération.

```
db.fish.find({}).limit(2).explain("executionStats");
```

Le résultat de cette opération ressemble à ceci.

```
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "test.fish",
        "winningPlan" : {
            "stage" : "SUBSCAN",
            "inputStage" : {
                "stage" : "LIMIT_SKIP",
                "inputStage" : {
                    "stage" : "COLLSCAN"
                }
            }
        }
    },
    "executionStats" : {
        "executionSuccess" : true,
        "executionTimeMillis" : "0.063",
        "planningTimeMillis" : "0.040",
        "executionStages" : {
            "stage" : "SUBSCAN",
            "nReturned" : "2",
            "executionTimeMillisEstimate" : "0.012",
            "inputStage" : {
                "stage" : "LIMIT_SKIP",
                "nReturned" : "2",
                "executionTimeMillisEstimate" : "0.005",
                "inputStage" : {
                    "stage" : "COLLSCAN",
                    "nReturned" : "2",
                    "executionTimeMillisEstimate" : "0.005"
                }
            }
        }
    },
    "serverInfo" : {
```

```
        "host" : "enginedemo",
        "port" : 27017,
        "version" : "3.6.0"
    },
    "ok" : 1
}
```

Si vous êtes intéressé uniquement par les informations relatives à `executionStats` de la requête ci-dessus, vous pouvez utiliser la commande suivante. Pour les petites collections, le processeur de requêtes Amazon DocumentDB peut choisir de ne pas utiliser d'index si les gains de performances sont négligeables.

```
db.fish.find({}).limit(2).explain("executionStats").executionStats;
```

Cache du plan de requêtes

Afin d'optimiser les performances et de réduire la durée de planification, Amazon DocumentDB met en cache les plans de requêtes en interne. Cela permet d'exécuter des requêtes de même forme directement à l'aide d'un plan mis en cache.

Cependant, cette mise en cache peut parfois entraîner un retard aléatoire pour la même requête ; par exemple, une requête dont l'exécution prend généralement une seconde peut parfois prendre dix secondes. En effet, au fil du temps, l'instance du lecteur a mis en cache différentes formes de la requête, consommant ainsi de la mémoire. Si vous rencontrez cette lenteur aléatoire, aucune action n'est nécessaire pour libérer la mémoire : le système gérera l'utilisation de la mémoire à votre place et une fois que la mémoire aura atteint un certain seuil, elle sera automatiquement libérée.

Expliquer les résultats

Si vous souhaitez renvoyer des informations sur les plans de requête, Amazon DocumentDB prend en charge le mode verbosité. `queryPlanner` Les `explain` résultats renvoient le plan de requête sélectionné par l'optimiseur dans un format similaire au suivant :

```
{
   "queryPlanner" : {
      "plannerVersion" : <int>,
      "namespace" : <string>,
```

```
      "winningPlan" : {
         "stage" : <STAGE1>,
         ...
         "inputStage" : {
            "stage" : <STAGE2>,
            ...
            "inputStage" : {
               ...
            }
         }
      }
   }
}
```

Les sections suivantes définiront les `explain` résultats communs.

Rubriques

- Étape de numérisation et de filtrage
- Intersection de l'index
- Union indicielle
- Intersection/union à indices multiples
- Indice composé
- Étape de tri
- Phase de groupes

Étape de numérisation et de filtrage

L'optimiseur peut choisir l'un des scans suivants :

COLLSCAN

Cette étape est une analyse de collecte séquentielle.

```
{
    "stage" : "COLLSCAN"
}
```

IXSCAN

Cette étape analyse les clés d'index. L'optimiseur peut récupérer le document au cours de cette étape, ce qui peut entraîner l'ajout d'une étape FETCH ultérieurement.

```
db.foo.find({"a": 1})
{
    "stage" : "IXSCAN",
    "direction" : "forward",
    "indexName" : <idx_name>
}
```

FETCH

Si l'optimiseur a extrait des documents dans une étape autre que IXSCAN, le résultat inclura une étape FETCH. Par exemple, la requête IXSCAN ci-dessus peut entraîner une combinaison des étapes FETCH et IXSCAN :

```
db.foo.find({"a": 1})
{
    "stage" : "FETCH",
    "inputStage" : {
        "stage" : "IXSCAN",
        "indexName" : <idx_name>
    }
}
```

IXONLYSCAN analyse uniquement la clé d'index. Créer des index composés n'évitera pas FETCH.

Intersection de l'index

MIXEUR

Amazon DocumentDB peut inclure un stage IXAND avec un tableau InputStages d'IXSCAN s'il peut utiliser l'intersection d'index. Par exemple, nous pouvons voir des résultats tels que :

```
{
    "stage" : "FETCH",
    "inputStage" : {
```

```
        "stage" : "IXAND",
        "inputStages" : [
            {
                "stage" : "IXSCAN",
                "indexName" : "a_1"
            },
            {
                "stage" : "IXSCAN",
                "indexName" : "b_1"
            }
        ]
    }
}
```

Union indicielle

IXOR

Comme pour l'intersection d'index, Amazon DocumentDB peut inclure un `IXOR` stage avec un `inputStages` tableau pour l'`$or`opérateur.

```
db.foo.find({"$or": [{"a": {"$gt": 2}}, {"b": {"$lt": 2}}]})
```

Pour la requête ci-dessus, la sortie d'explication peut ressembler à ceci :

```
{
    "stage" : "FETCH",
    "inputStage" : {
        "stage" : "IXOR",
        "inputStages" : [
            {
                "stage" : "IXSCAN",
                "indexName" : "a_1"
            },
            {
                "stage" : "IXSCAN",
                "indexName" : "b_1"
            }
        ]
    }
}
```

Intersection/union à indices multiples

Amazon DocumentDB peut combiner plusieurs étapes d'intersection ou d'union d'index, puis récupérer le résultat. Par exemple :

```
{
    "stage" : "FETCH",
    "inputStage" : {
        "stage" : "IXOR",
        "inputStages" : [
            {
                "stage" : "IXSCAN",
                ...
            },
            {
                "stage" : "IXAND",
                "inputStages" : [
                    {
                        "stage" : "IXSCAN",
                        ...
                    },
                    {
                        "stage" : "IXSCAN",
                        ...
                    }
                ]
            }
        ]
    }
}
```

L'utilisation des étapes d'intersection ou d'union d'index n'est pas affectée par le type d'indice (épars, composé, etc.).

Indice composé

L'utilisation de l'index composé Amazon DocumentDB n'est pas limitée dans les premiers sous-ensembles de champs indexés ; elle peut utiliser l'index avec le suffixe, mais cela peut ne pas être très efficace.

Par exemple, l'index composé de `{ a: 1, b: -1 }` peut prendre en charge les trois requêtes ci-dessous :

```
db.orders.find( { a: 1 } } )
```

```
db.orders.find( { b: 1 } } )
```

```
db.orders.find( { a: 1, b: 1 } } )
```

Étape de tri

S'il existe un index sur la ou les clés de tri demandées, Amazon DocumentDB peut utiliser cet index pour obtenir la commande. Dans ce cas, le résultat n'inclura pas d'`SORT`étape, mais plutôt une `IXSCAN` étape. Si l'optimiseur privilégie un tri simple, il inclura une étape comme celle-ci :

```
{
    "stage" : "SORT",
    "sortPattern" : {
        "a" : 1,
        "b" : -1
    }
}
```

Phase de groupes

Amazon DocumentDB prend en charge deux stratégies de groupe différentes :

- `SORT_AGGREGATE`: agrégat de tri sur disque.
- `HASH_AGGREGATE`: agrégat de hachage en mémoire.

Interrogation de données géospatiales avec Amazon DocumentDB

Cette section explique comment interroger des données géospatiales avec Amazon DocumentDB. Après avoir lu cette section, vous serez en mesure de savoir comment stocker, interroger et indexer des données géospatiales dans Amazon DocumentDB.

Rubriques

Présentation

Les cas d'utilisation courants de la géospatiale impliquent l'analyse de proximité à partir de vos données. Par exemple, « trouver tous les aéroports situés dans un rayon de 80 miles de Seattle » ou « trouver les restaurants les plus proches d'un endroit donné ». Amazon DocumentDB utilise la spécification GeoJSON pour représenter les données géospatiales. GeoJSON est une spécification open source pour le formatage JSON des formes dans un espace de coordonnées. Les coordonnées GeoJSON capturent à la fois la longitude et la latitude, représentant les positions sur une sphère semblable à la Terre.

Indexation et stockage de données géospatiales

Amazon DocumentDB utilise le type GeoJSON « Point » pour stocker les données géospatiales. Chaque document (ou sous-document) GeoJSON est généralement composé de deux champs :

- type : la forme représentée, qui indique à Amazon DocumentDB comment interpréter le champ « coordonnées ». Pour le moment, Amazon DocumentDB ne prend en charge que les points
- coordonnées — une paire de latitude et de longitude représentée sous la forme d'un objet dans un tableau — [longitude, latitude]

Amazon DocumentDB utilise également des index 2dsphere pour indexer les données géospatiales. Amazon DocumentDB prend en charge les points d'indexation. Amazon DocumentDB prend en charge les requêtes de proximité avec l'indexation 2dsphere.

Imaginons un scénario dans lequel vous créez une application pour un service de livraison de nourriture. Vous souhaitez stocker la paire de latitudes et de longitude de différents restaurants dans Amazon DocumentDB. Pour ce faire, nous vous recommandons de créer d'abord un index dans le champ Géospatial contenant la paire de latitude et de longitude.

```
use restaurantsdb
db.usarestaurants.createIndex({location:"2dsphere"})
```

Le résultat de cette commande ressemblerait à ceci :

```
{
```

```
 "createdCollectionAutomatically" : true,
 "numIndexesBefore" : 1,
 "numIndexesAfter" : 2,
 "ok" : 1
}
```

Une fois que vous avez créé un index, vous pouvez commencer à insérer des données dans votre collection Amazon DocumentDB.

```
db.usarestaurants.insert({
   "state":"Washington",
   "city":"Seattle",
   "name":"Thai Palace",
   "rating": 4.8,
   "location":{
      "type":"Point",
      "coordinates":[
         -122.3264,
         47.6009
      ]
   }
});

db.usarestaurants.insert({
   "state":"Washington",
   "city":"Seattle",
   "name":"Noodle House",
   "rating": 4.8,
   "location":{
      "type":"Point",
      "coordinates":[
        -122.3517,
         47.6159
      ]
   }
});

db.usarestaurants.insert({
   "state":"Washington",
   "city":"Seattle",
   "name":"Curry House",
   "rating": 4.8,
   "location":{
```

```
      "type":"Point",
      "coordinates":[
         -121.4517,
         47.6229
      ]
   }
});
```

Interrogation de données géospatiales

Amazon DocumentDB prend en charge les requêtes de proximité, d'inclusion et d'intersection de données géospatiales. Un bon exemple de requête de proximité consiste à rechercher tous les points (tous les aéroports) situés à moins d'une certaine distance et à plus d'une distance d'un autre point (ville). Un bon exemple de requête d'inclusion consiste à rechercher tous les points (tous les aéroports) situés dans une zone/un polygone spécifique (État de New York). Un bon exemple de requête d'intersection consiste à trouver un polygone (état) qui croise un point (ville). Vous pouvez utiliser les opérateurs géospatiaux suivants pour obtenir des informations à partir de vos données.

- **`$nearSphere`**- `$nearSphere` est un opérateur de recherche qui permet de rechercher des points du plus proche au plus éloigné d'un point GeoJSON.
- **`$geoNear`**- `$geoNear` est un opérateur d'agrégation qui permet de calculer la distance en mètres à partir d'un point GeoJSON.
- **`$minDistance`**- `$minDistance` est un opérateur de recherche utilisé conjointement avec `$nearSphere` ou `$geoNear` pour filtrer les documents situés au moins à la distance minimale spécifiée par rapport au point central.
- **`$maxDistance`**- `$maxDistance` est un opérateur de recherche utilisé conjointement avec `$nearSphere` ou `$geoNear` pour filtrer les documents situés au maximum à la distance maximale spécifiée par rapport au point central.
- **`$geoWithin`**- `$geoWithin` est un opérateur de recherche qui permet de rechercher des documents contenant des données géospatiales qui existent entièrement dans une forme spécifiée, telle qu'un polygone.
- **`$geoIntersects`**- `$geoIntersects` est un opérateur de recherche qui permet de rechercher des documents dont les données géospatiales croisent un objet GeoJSON spécifié.

> **Note**
>
> `$geoNear`et `$nearSphere` exigez un index 2dsphere sur le champ GeoJSON que vous utilisez dans votre requête de proximité.

Exemple 1

Dans cet exemple, vous allez apprendre comment rechercher tous les restaurants (points) triés par distance la plus proche d'une adresse (point).

Pour effectuer une telle requête, vous pouvez l'utiliser `$geoNear` pour calculer la distance d'un ensemble de points par rapport à un autre point. Vous pouvez également ajouter le `distanceMultiplier` pour mesurer la distance en kilomètres.

```
db.usarestaurants.aggregate([
   {
      "$geoNear":{
         "near":{
            "type":"Point",
            "coordinates":[
               -122.3516,
               47.6156
            ]
         },
         "spherical":true,
         "distanceField":"DistanceKilometers",
         "distanceMultiplier":0.001
      }
   }
])
```

La commande ci-dessus renverrait les restaurants triés par distance (du plus proche au plus éloigné) du point spécifié. La sortie de cette commande ressemblerait à ceci

```
{ "_id" : ObjectId("611f3da985009a81ad38e74b"), "state" : "Washington", "city" :
 "Seattle", "name" : "Noodle House", "rating" : 4.8, "location" : { "type" : "Point",
 "coordinates" : [ -122.3517, 47.6159 ] }, "DistanceKilometers" : 0.03422834547294996 }
{ "_id" : ObjectId("611f3da185009a81ad38e74a"), "state" : "Washington", "city" :
 "Seattle", "name" : "Thai Palace", "rating" : 4.8, "location" : { "type" : "Point",
 "coordinates" : [ -122.3264, 47.6009 ] }, "DistanceKilometers" : 2.5009390081704277 }
```

```
{ "_id" : ObjectId("611f3dae85009a81ad38e74c"), "state" : "Washington", "city" :
 "Seattle", "name" : "Curry House", "rating" : 4.8, "location" : { "type" : "Point",
 "coordinates" : [ -121.4517, 47.6229 ] }, "DistanceKilometers" : 67.52845344856914 }
```

Pour limiter le nombre de résultats d'une requête, utilisez l'numoption `limit` ou.

`limit`:

```
db.usarestaurants.aggregate([
   {
      "$geoNear":{
         "near":{
            "type":"Point",
            "coordinates":[
               -122.3516,
               47.6156
            ]
         },
         "spherical":true,
         "distanceField":"DistanceKilometers",
         "distanceMultiplier":0.001,
         "limit": 10
      }
   }
])
```

num:

```
db.usarestaurants.aggregate([
   {
      "$geoNear":{
         "near":{
            "type":"Point",
            "coordinates":[
               -122.3516,
               47.6156
            ]
         },
         "spherical":true,
         "distanceField":"DistanceKilometers",
         "distanceMultiplier":0.001,
         "num": 10
      }
```

```
   }
])
```

> **Note**
>
> `$geoNear`stage prend en charge les `num` options `limit` et pour spécifier le nombre maximum de documents à renvoyer. `$geoNear`renvoie un maximum de 100 documents par défaut si les `num` options `limit` ou ne sont pas spécifiées. Cette valeur est remplacée par la valeur de l'`$limit`étape si elle est présente et la valeur est inférieure à 100.

Exemple 2

Dans cet exemple, vous allez apprendre comment trouver tous les restaurants (points) situés dans un rayon de 2 kilomètres d'une adresse spécifique (point). Pour effectuer une telle requête, vous pouvez utiliser un minimum `$nearSphere` `$minDistance` et un maximum `$maxDistance` à partir d'un point GeoJSON

```
db.usarestaurants.find({
   "location":{
      "$nearSphere":{
         "$geometry":{
            "type":"Point",
            "coordinates":[
               -122.3516,
               47.6156
            ]
         },
         "$minDistance":1,
         "$maxDistance":2000
      }
   }
},
{
   "name":1
})
```

La commande ci-dessus renverrait les restaurants à une distance maximale de 2 kilomètres du point spécifié. La sortie de cette commande ressemblerait à ceci

```
{ "_id" : ObjectId("611f3da985009a81ad38e74b"), "name" : "Noodle House" }
```

Limites

Amazon DocumentDB ne prend pas en charge l'interrogation ou l'indexation des polygones,,,, LineString et. MultiPoint MultiPolygon MultiLineString GeometryCollection

Index partiel

Un index partiel indexe les documents d'une collection qui répond à un critère de filtre spécifié. La fonctionnalité d'index partiel est prise en charge dans les clusters basés sur des instances Amazon DocumentDB 5.0.

Rubriques

- Création d'un index partiel
- Opérateurs pris en charge
- Requête utilisant un index partiel
- Fonctionnalités de l'index partiel
- Limitations partielles de l'indice

Création d'un index partiel

Pour créer un index partiel, utilisez la `createIndex()` méthode avec l'`partialFilterExpression`option. Par exemple, l'opération suivante crée un index composé unique dans la collection de commandes qui indexe les documents ayant un `OrderID` et dont le `isDelivered` champ est vrai :

```
db.orders.createIndex(
  {"category": 1, "CustomerId": 1, "OrderId": 1},
  {"unique": true, "partialFilterExpression":
    {"$and": [
      {"OrderId": {"$exists": true}},
      {"isDelivered": {"$eq": false}}
    ]}
  }
)
```

Opérateurs pris en charge

- $eq
- $exists
- $and (uniquement au niveau supérieur)
- $gt/$gte/$lt/$lte (le scan d'index n'est utilisé que lorsque le filtre, défini dans la requête, correspond exactement à l'expression du filtre partiel) (voir Limitations)

Requête utilisant un index partiel

Les modèles de requête suivants sont possibles à l'aide d'index partiels :

- Le prédicat de requête correspond exactement à l'expression du filtre d'index partiel :

```
db.orders.find({"$and": [
    {"OrderId": {"$exists": true}},
    {"isDelivered": {"$eq": false}}
  ]}).explain()
```

- Le résultat attendu du filtre de requête est un sous-ensemble logique du filtre partiel :

```
db.orders.find({"$and": [
    {"OrderId": {"$exists": true}},
    {"isDelivered": {"$eq": false}},
    {"OrderAmount": {"$eq": "5"}}
  ]}).explain()
```

- Un sous-prédicat de la requête peut être utilisé conjointement avec d'autres index :

```
db.orders.createIndex({"anotherIndex":1})
db.orders.find({ "$or": [
      {"$and": [
        {"OrderId": {"$exists": true}},
        {"isDelivered": {"$eq": false}}
      ]},
      {"anotherIndex": {"$eq": 5}}
    ]
  }).explain()
```

> **Note**
>
> Un planificateur de requêtes peut choisir d'utiliser une analyse de collection plutôt qu'une analyse d'index s'il est efficace de le faire. Cela se produit généralement pour de très petites collections ou des requêtes qui renvoient une grande partie d'une collection.

Fonctionnalités de l'index partiel

Répertorier les index partiels

Répertoriez les index partiels à partialFilterExpression l'aide de l'`getIndex`opération. Par exemple, l'`getIndex`opération émise dans répertorie les index partiels avec les champs key, name et PartialFilterExpressions :

```
db.orders.getIndexes()
```

Cet exemple renvoie le résultat suivant :

```
[
  {
    "v" : 4,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "ecommerceApp.orders"
  },
  {
    "v" : 4,
    "unique" : true,
    "key" : {
      "category" : 1,
      "" : 1,
      "CustomerId" : 1,
      "OrderId" : 1
    },
    "name" : "category_1_CustID_1_OrderId_1",
    "ns" : "ecommerceApp.orders",
    "partialFilterExpression" : {
      "$and" : [
```

```
        {"OrderId": {"$exists": true}},
        {"isDelivered": {"$eq": false}}
      ]
    }
  }
]
```

Expression de filtre partiel multiple sur la même clé:order

Différents index partiels peuvent être créés pour les mêmes combinaisons de champs (key:order). Ces index doivent porter un nom différent.

```
db.orders.createIndex(
  {"OrderId":1},
  {
    name:"firstPartialIndex",
    partialFilterExpression:{"OrderId":{"$exists": true}}
  }
)
```

```
db.orders.createIndex(
  {"OrderId":1},
  {
    name:"secondPartialIndex",
    partialFilterExpression:{"OrderId":{"$gt": 1000}}
  }
)
```

Exécutez `getIndexes` l'opération pour répertorier tous les index de la collection :

```
db.orders.getIndexes()
```

Ces exemples renvoient le résultat suivant :

```
[
  {
    "v" : 4,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "ecommerceApp.orders"
```

```
  },
  {
    "v" : 4,
    "key" : {
      "OrderId" : 1
    },
    "name" : "firstPartialIndex",
    "ns" : "ecommerceApp.orders",
    "partialFilterExpression" : {"OrderId":{"$exists": true}}
  },
  {
    "v" : 4,
    "key" : {
      "OrderId" : 1
    },
    "name" : "secondPartialIndex",
    "ns" : "ecommerceApp.orders",
    "partialFilterExpression" : {"OrderId":{"$gt": 1000}}
  }
]
```

> **Important**
>
> Les noms d'index doivent être différents et ne doivent être supprimés que par leur nom.

Index avec propriétés partielles et TTL

Vous pouvez également créer des index dotés de propriétés partielles et TTL en spécifiant à la fois les deux `partialFilterExpression` et les `expireAfterSeconds` options lors de la création de l'index. Cela vous permet de mieux contrôler les documents qui sont désormais supprimés d'une collection.

Par exemple, il se peut que vous disposiez d'un index TTL qui identifie les documents à supprimer après un certain laps de temps. Vous pouvez désormais définir des conditions supplémentaires concernant le moment où vous devez supprimer des documents à l'aide de l'option d'indexation partielle :

```
db.orders.createIndex(
    { "OrderTimestamp": 1 },
    {
```

```
        expireAfterSeconds: 3600 ,
        partialFilterExpression: { "isDelivered": { $eq: true } }
    }
)
```

Cet exemple renvoie le résultat suivant :

```
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1,
        "operationTime" : Timestamp(1234567890, 1)
}
```

Exécutez l'`getIndexes`opération pour répertorier les index présents dans la collection :

```
db.orders.getIndexes()
[
    {
        "v" : 4,
        "key" : {
            "_id" : 1
        },
        "name" : "_id_",
        "ns" : "test.orders"
    }
```

Cet exemple renvoie le résultat suivant :

```
[
    {
        "v": 4,
        "key": {
            "_id": 1
        },
        "name": "_id_",
        "ns": "ecommerceApp.orders"
    },
    {
        "v": 4,
        "key": {
```

```
            "OrderTimestamp": 1
        },
        "name": "OrderTimestamp_1",
        "ns": "ecommerceApp.orders",
        "partialFilterExpression": {
            "isDelivered": {
                "$eq": true
            }
        },
        "expireAfterSeconds": 3600
    }
]
```

Limitations partielles de l'indice

Les limites suivantes s'appliquent à la fonction d'index partiel :

- Les requêtes d'inégalité dans Amazon DocumentDB n'utiliseront un index partiel que lorsque le prédicat du filtre de requêtes correspond exactement au même type de données `partialFilterExpression` et qu'il est du même type de données.

  > **Note**
  >
  > `$hint`Ne peut même pas être utilisé pour forcer IXSCAN dans le cas ci-dessus.

  Dans l'exemple suivant, le `partialFilterExpression` est uniquement appliqué `field1` mais pas `field2` :

  ```
  db.orders.createIndex(
    {"OrderAmount": 1},
    {"partialFilterExpression": { OrderAmount : {"$gt" : 5}}}
  )

  db.orders.find({OrderAmount : {"$gt" : 5}}) // Will use partial index
  db.orders.find({OrderAmount : {"$gt" : 6}}) // Will not use partial index
  db.orders.find({OrderAmount : {"$gt" : Decimal128(5.00)}}) // Will not use partial
   index
  ```

- A `partialFilterExpression` avec opérateurs de tableau ne sont pas pris en charge. L'opération suivante va générer une erreur :

```
db.orders.createIndex(
  {"CustomerId":1},
  {'partialFilterExpression': {'OrderId': {'$eq': [1000, 1001, 1002]}}}
)
```

- Les opérateurs suivants ne sont pas pris en charge partialFilterExpression sur le terrain :
  - `$all`(opérateur de tableau)
  - `$mod`(opérateur de tableau)
  - `$or`
  - `$xor`
  - `$not`
  - `$nor`
- Le type de données de l'expression du filtre et celui du filtre doivent être identiques.

Effectuer une recherche de texte avec Amazon DocumentDB

La fonction native de recherche en texte intégral d'Amazon DocumentDB vous permet d'effectuer une recherche textuelle sur de grands ensembles de données textuelles à l'aide d'index de texte spécifiques. Cette section décrit les fonctionnalités de la fonctionnalité d'index de texte et explique comment créer et utiliser des index de texte dans Amazon DocumentDB. Les limites de recherche de texte sont également répertoriées.

Rubriques

Fonctionnalités prises en charge

La recherche de texte Amazon DocumentDB prend en charge les fonctionnalités compatibles avec l'API MongoDB suivantes :

- Créez des index de texte sur un seul champ.
- Créez des index de texte composés qui incluent plusieurs champs de texte.
- Effectuez des recherches portant sur un seul mot ou sur plusieurs mots.
- Contrôlez les résultats de recherche à l'aide de pondérations.
- Triez les résultats de recherche par score.
- Utilisez l'index de texte dans le pipeline d'agrégation.
- Recherchez la phrase exacte.

Utilisation de l'index de texte Amazon DocumentDB

Pour créer un index de texte sur un champ contenant des données sous forme de chaîne, spécifiez la chaîne « texte » comme indiqué ci-dessous :

Index à champ unique :

```
db.test.createIndex({"comments": "text"})
```

Cet index prend en charge les requêtes de recherche de texte dans le champ de chaîne « commentaires » de la collection spécifiée.

Créez un index de texte composé sur plusieurs champs de chaîne :

```
db.test.createIndex({"comments": "text", "title":"text"})
```

Cet index prend en charge les requêtes de recherche de texte dans les champs de chaîne « commentaires » et « titre » de la collection spécifiée. Vous pouvez spécifier jusqu'à 30 champs lors de la création d'un index de texte composé. Une fois créées, vos requêtes de recherche de texte interrogeront tous les champs indexés.

> **Note**
>
> Un seul index de texte est autorisé par collection.

Répertorier un index de texte dans une collection Amazon DocumentDB

Vous pouvez utiliser `getIndexes()` votre collection pour identifier et décrire les index, y compris les index textuels, comme indiqué dans l'exemple ci-dessous :

```
rs0:PRIMARY> db.test.getIndexes()
[
   {
      "v" : 4,
      "key" : {
         "_id" : 1
      },
      "name" : "_id_",
      "ns" : "test.test"
   },
   {
      "v" : 1,
      "key" : {
         "_fts" : "text",
         "_ftsx" : 1
      },
      "name" : "contents_text",
      "ns" : "test.test",
      "default_language" : "english",
      "weights" : {
         "comments" : 1
      },
      "textIndexVersion" : 1
   }
]
```

Une fois que vous avez créé un index, commencez à insérer des données dans votre collection Amazon DocumentDB.

```
db.test.insertMany([{"_id": 1, "star_rating": 4, "comments": "apple is red"},
                    {"_id": 2, "star_rating": 5, "comments": "pie is delicious"},
                    {"_id": 3, "star_rating": 3, "comments": "apples, oranges - healthy
 fruit"},
                    {"_id": 4, "star_rating": 2, "comments": "bake the apple pie in the
 oven"},
                    {"_id": 5, "star_rating": 5, "comments": "interesting couch"},
```

```
                    {"_id": 6, "star_rating": 5, "comments": "interested in couch for
 sale, year 2022"}])
```

Exécution de requêtes de recherche de texte

Exécuter une requête de recherche sous forme de texte contenant un seul mot

Vous devrez utiliser les `$search` opérateurs `$text` and pour effectuer des recherches de texte. L'exemple suivant renvoie tous les documents dans lesquels votre champ de texte indexé contient la chaîne « apple » ou « apple » dans d'autres formats tels que « apples » :

```
db.test.find({$text: {$search: "apple"}})
```

Sortie :

Le résultat de cette commande ressemble à ceci :

```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red" }
{ "_id" : 3, "star_rating" : 3, "comments" : "apples, oranges - healthy fruit" }
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
```

Lancer une recherche textuelle comportant plusieurs mots

Vous pouvez également effectuer des recherches textuelles de plusieurs mots sur vos données Amazon DocumentDB. La commande ci-dessous renvoie des documents dont le champ de texte indexé contient « apple » ou « pie » :

```
db.test.find({$text: {$search: "apple pie"}})
```

Sortie :

Le résultat de cette commande ressemble à ceci :

```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red" }
{ "_id" : 2, "star_rating" : 5, "comments" : "pie is delicious" }
{ "_id" : 3, "star_rating" : 3, "comments" : "apples, oranges - healthy fruit" }
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
```

Lancer une recherche textuelle de plusieurs mots

Pour une recherche de phrases comportant plusieurs mots, utilisez cet exemple :

```
db.test.find({$text: {$search: "\"apple pie\""}})
```

Sortie :

La commande ci-dessus renvoie des documents dont le champ de texte indexé contient l'expression exacte « tarte aux pommes ». Le résultat de cette commande ressemble à ceci :

```
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
```

Lancer une recherche de texte à l'aide de filtres

Vous pouvez également associer la recherche textuelle à d'autres opérateurs de requête pour filtrer les résultats en fonction de critères supplémentaires :

```
db.test.find({$and: [{star_rating: 5}, {$text: {$search: "interest"}}]})
```

Sortie :

La commande ci-dessus renvoie des documents contenant un champ de texte indexé contenant n'importe quelle forme d' « intérêt » et un « star_rating » égal à 5. Le résultat de cette commande ressemble à ceci :

```
{ "_id" : 5, "star_rating" : 5, "comments" : "interesting couch" }
{ "_id" : 6, "star_rating" : 5, "comments" : "interested in couch for sale, year
 2022" }
```

Limiter le nombre de documents renvoyés lors d'une recherche textuelle

Vous pouvez choisir de limiter le nombre de documents renvoyés en utilisant `limit` :

```
db.test.find({$and: [{star_rating: 5}, {$text: {$search: "couch"}}]}).limit(1)
```

Sortie :

La commande ci-dessus renvoie un résultat satisfaisant au filtre :

```
{ "_id" : 5, "star_rating" : 5, "comments" : "interesting couch" }
```

Trier les résultats par score textuel

L'exemple suivant trie les résultats de recherche de texte par score de texte :

```
db.test.find({$text: {$search: "apple"}}, {score: {$meta: "textScore"}}).sort({score:
 {$meta: "textScore"}})
```

Sortie :

La commande ci-dessus renvoie des documents contenant un champ de texte indexé contenant « apple » ou « apple » dans d'autres formats tels que « apple », et trie le résultat en fonction de la pertinence du document par rapport au terme de recherche. Le résultat de cette commande ressemble à ceci :

```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red", "score" :
 0.6079270860936958 }
{ "_id" : 3, "star_rating" : 3, "comments" : "apples, oranges - healthy fruit",
 "score" : 0.6079270860936958 }
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven",
 "score" : 0.6079270860936958 }
```

`$text`et `$search` sont également pris en charge pour les `delete` commandes `aggregate` `count` `findAndModify`,`update`,, et.

Opérateurs d'agrégation

Pipeline d'agrégation utilisant **`$match`**

```
db.test.aggregate(
   [{ $match: { $text: { $search: "apple pie" } } }]
)
```

Sortie :

La commande ci-dessus renvoie les résultats suivants :

```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red" }
{ "_id" : 3, "star_rating" : 3, "comments" : "apple - a healthy fruit" }
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
```

```
{ "_id" : 2, "star_rating" : 5, "comments" : "pie is delicious" }
```

Combinaison d'autres opérateurs d'agrégation

```
db.test.aggregate(
   [
      { $match: { $text: { $search: "apple pie" } } },
      { $sort: { score: { $meta: "textScore" } } },
      { $project: { score: { $meta: "textScore" } } }
   ]
)
```

Sortie :

La commande ci-dessus renvoie les résultats suivants :

```
{ "_id" : 4, "score" : 0.6079270860936958 }
{ "_id" : 1, "score" : 0.3039635430468479 }
{ "_id" : 2, "score" : 0.3039635430468479 }
{ "_id" : 3, "score" : 0.3039635430468479 }
```

Spécifier plusieurs champs lors de la création d'un index de texte

Vous pouvez attribuer des pondérations à un maximum de trois champs dans votre index de texte composé. Le poids par défaut attribué à un champ dans un index de texte est de un (1). Le poids est un paramètre facultatif et doit être compris entre 1 et 100 000.

```
db.test.createIndex(
   {
     "firstname": "text",
     "lastname": "text",
     ...
   },
   {
     weights: {
       "firstname": 5,
       "lastname":10,
       ...
     },
     name: "name_text_index"
   }
```

```
)
```

Différences avec MongoDB

La fonctionnalité d'index de texte d'Amazon DocumentDB utilise un index inversé avec un algorithme de périodicité. Les index de texte sont épars par défaut. En raison des différences entre la logique d'analyse, les délimiteurs de tokenisation, etc., le même jeu de résultats que MongoDB peut ne pas être renvoyé pour le même ensemble de données ou la même forme de requête.

Les différences supplémentaires suivantes existent entre l'index de texte Amazon DocumentDB et MongoDB :

- Les index composés utilisant des index non textuels ne sont pas pris en charge.
- Les index de texte Amazon DocumentDB ne distinguent pas les majuscules des minuscules et les signes diacritiques.
- Seule la langue anglaise est prise en charge avec l'index de texte.
- L'indexation du texte des champs matriciels (ou multiclés) n'est pas prise en charge. Par exemple, la création d'un index de texte sur « a » avec le document {« a » : [« apple », « pie »]} échouera.
- L'indexation de texte générique n'est pas prise en charge.
- Les index de texte uniques ne sont pas pris en charge.
- L'exclusion d'un terme n'est pas prise en charge.

Bonnes pratiques et directives

- Pour des performances optimales sur les requêtes de recherche de texte impliquant un tri par score de texte, nous vous recommandons de créer l'index de texte avant de charger les données.
- Les index de texte nécessitent un espace de stockage supplémentaire pour une copie interne optimisée des données indexées. Cela entraîne des coûts supplémentaires.

Limites

La recherche de texte présente les limites suivantes dans Amazon DocumentDB :

- La recherche de texte est prise en charge uniquement sur les clusters basés sur des instances Amazon DocumentDB 5.0.

Résolution des problèmes liés à Amazon DocumentDB

Les sections suivantes fournissent des informations sur la manière de résoudre les problèmes que vous pourriez rencontrer lors de l'utilisation d'Amazon DocumentDB (avec compatibilité avec MongoDB).

Rubriques

Problèmes de connexion

Vous rencontrez des difficultés pour vous connecter ? Voici quelques scénarios courants et comment les résoudre.

Rubriques

Impossible de se connecter à un point de terminaison Amazon DocumentDB

Lorsque vous essayez de vous connecter à Amazon DocumentDB, voici l'un des messages d'erreur les plus courants que vous pouvez recevoir.

```
connecting to: mongodb://docdb-2018-11-08-21-47-27.cluster-ccuszbx3pn5e.us-east-
1.docdb.amazonaws.com:27017/
2018-11-14T14:33:46.451-0800 W NETWORK [thread1] Failed to connect to
172.31.91.193:27017 after 5000ms milliseconds, giving up.
2018-11-14T14:33:46.452-0800 E QUERY [thread1] Error: couldn't connect to server
docdb-2018-11-08-21-47-27.cluster-ccuszbx3pn5e.us-east-1.docdb.amazonaws.com:27017,
```

```
connection attempt failed :
connect@src/mongo/shell/mongo.js:237:13
@(connect):1:6
exception: connect failed
```

Ce message d'erreur signifie généralement que votre client (le shell mongo dans cet exemple) ne peut pas accéder au point de terminaison Amazon DocumentDB. Cela peut être le cas pour plusieurs raisons :

Rubriques

- Connexion à partir de points de terminaison publics
- Connexions interrégionales
- Connexion depuis différents Amazon VPC
- Le groupe de sécurité bloque les connexions entrantes
- Problème de préférence de lecture du pilote Java Mongo

Connexion à partir de points de terminaison publics

Vous essayez de vous connecter à un cluster Amazon DocumentDB directement depuis votre ordinateur portable ou votre machine de développement locale.

Toute tentative de connexion à un cluster Amazon DocumentDB directement depuis un point de terminaison public, tel que votre ordinateur portable ou votre machine de développement locale, échouera. Amazon DocumentDB est uniquement destiné au cloud privé virtuel (VPC) et ne prend actuellement pas en charge les points de terminaison publics. Ainsi, vous ne pouvez pas vous connecter directement à votre cluster Amazon DocumentDB depuis votre ordinateur portable ou depuis un environnement de développement local en dehors de votre VPC.

Pour vous connecter à un cluster Amazon DocumentDB depuis l'extérieur d'un Amazon VPC, vous pouvez utiliser un tunnel SSH. Pour plus d’informations, consultez Connexion à un cluster Amazon DocumentDB depuis l'extérieur d'un Amazon VPC. En outre, si votre environnement de développement se trouve dans un autre Amazon VPC, vous pouvez également utiliser VPC Peering et vous connecter à votre cluster Amazon DocumentDB depuis un autre Amazon VPC de la même région ou d'une région différente.

Connexions interrégionales

Vous essayez de vous connecter à un cluster Amazon DocumentDB dans une autre région.

Si vous essayez de vous connecter à un cluster Amazon DocumentDB depuis une instance Amazon EC2 située dans une région autre que celle du cluster, par exemple en essayant de vous connecter à un cluster situé dans la région USA Est (Virginie du Nord) (us-east-1) depuis la région USA Ouest (Oregon) (us-west-2), la connexion échouera.

Pour vérifier la région de votre cluster Amazon DocumentDB, exécutez la commande suivante. La région est dans le point de terminaison.

```
aws docdb describe-db-clusters \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].Endpoint'
```

Le résultat de cette opération ressemble à ceci.

```
[
    "sample-cluster.node.us-east-1.docdb.amazonaws.com"
]
```

Pour vérifier la région de votre instance EC2, exécutez la commande suivante :

```
 aws ec2 describe-instances \
     --query 'Reservations[*].Instances[*].Placement.AvailabilityZone'
```

Le résultat de cette opération ressemble à ceci.

```
[
    [
        "us-east-1a"
    ]
]
```

Connexion depuis différents Amazon VPC

Vous essayez de vous connecter à un cluster Amazon DocumentDB à partir d'un VPC différent de celui sur lequel votre cluster est déployé.

Si votre cluster Amazon DocumentDB et votre instance Amazon EC2 se trouvent dans le Région AWS même environnement, mais pas dans le même Amazon VPC, vous ne pouvez pas vous connecter directement à votre cluster Amazon DocumentDB à moins que le peering VPC ne soit activé entre les deux Amazon VPC.

Pour vérifier le VPC Amazon de votre instance Amazon DocumentDB, exécutez la commande suivante.

```
aws docdb describe-db-instances \
   --db-instance-identifier sample-instance \
   --query 'DBInstances[*].DBSubnetGroup.VpcId'
```

Pour vérifier le VPC Amazon de votre instance Amazon EC2, exécutez la commande suivante.

```
aws ec2 describe-instances \
   --query 'Reservations[*].Instances[*].VpcId'
```

Le groupe de sécurité bloque les connexions entrantes

Vous essayez de vous connecter à un cluster Amazon DocumentDB et le groupe de sécurité du cluster n'autorise pas les connexions entrantes sur le port du cluster (port par défaut : 27017).

Supposons que votre cluster Amazon DocumentDB et votre instance Amazon EC2 se trouvent tous deux dans la même région et dans le même Amazon VPC et qu'ils utilisent le même groupe de sécurité Amazon VPC. Si vous ne parvenez pas à vous connecter à votre cluster Amazon DocumentDB, cela est probablement dû au fait que le groupe de sécurité (c'est-à-dire le pare-feu) de votre cluster n'autorise pas les connexions entrantes sur le port que vous avez choisi pour votre cluster Amazon DocumentDB (le port par défaut est 27017).

Pour vérifier le port de votre cluster Amazon DocumentDB, exécutez la commande suivante.

```
aws docdb describe-db-clusters \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].[DBClusterIdentifier,Port]'
```

Pour obtenir le groupe de sécurité Amazon DocumentDB pour votre cluster, exécutez la commande suivante.

```
aws docdb describe-db-clusters \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].[VpcSecurityGroups[*],VpcSecurityGroupId]'
```

Pour vérifier les règles relatives au trafic entrant pour votre groupe de sécurité, consultez les rubriques suivantes de la documentation Amazon EC2 :

- Autorisation du trafic entrant pour vos instances Linux
- Autorisation du trafic entrant pour vos instances Windows

Problème de préférence de lecture du pilote Java Mongo

Les préférences de lecture des clients ne sont pas respectées et certains clients ne peuvent pas écrire sur Amazon DocumentDB après un basculement, sauf s'ils redémarrent.

Ce problème, découvert pour la première fois dans Java Mongo Driver 3.7.x, se produit lorsqu'un client établit une connexion à Amazon DocumentDB à l'aide de la méthode `MongoClientSettings` et, en particulier, lors du chaînage. `applyToClusterSettings` Les paramètres du MongoClient cluster peuvent être définis à l'aide de différentes méthodes, telles que `hosts()requiredReplicaSetName()`, et`mode()`.

Lorsque le client ne spécifie qu'un seul hôte dans la `hosts()` méthode, le mode est défini sur au `ClusterConnectionMode.SINGLE` lieu de `ClusterConnectionMode.MULTIPLE` Cela amène le client à ignorer la préférence de lecture et à se connecter uniquement au serveur configuré dans`hosts()`. Ainsi, même si les paramètres du client sont initialisés comme ci-dessous, toutes les lectures seront toujours dirigées vers le primaire plutôt que vers le secondaire.

```
final ServerAddress serverAddress0 = new ServerAddress("cluster-endpoint", 27317));
    final MongoCredential credential = MongoCredential.createCredential("xxx",
            "admin", "xxxx".toCharArray());
    final MongoClientSettings settings = MongoClientSettings.builder()
            .credential(credential)
            .readPreference(ReadPreference.secondaryPreferred())
            .retryWrites(false)
            .applyToSslSettings(builder -> builder
                    .enabled(false))
            .applyToClusterSettings(builder -> builder.hosts(
                            Arrays.asList(serverAddress0
                            ))
                    .requiredReplicaSetName("rs0"))
            .build();
    MongoClient mongoClient = MongoClients.create(settings);
```

Étui Failover

En utilisant les paramètres de connexion client ci-dessus, en cas de basculement et de mise à jour différée de l'enregistrement DNS pour le point de terminaison du rédacteur du cluster, le

client essaiera toujours d'envoyer des écritures à l'ancien enregistreur (devenu lecteur après le basculement). Cela entraîne une erreur côté serveur (et non principale) qui n'est pas gérée correctement par le pilote Java (cela fait toujours l'objet d'une enquête). Ainsi, le client peut être laissé dans un mauvais état jusqu'au redémarrage du serveur d'applications, par exemple.

Il existe deux solutions pour contourner ce problème :

- Les clients qui se connectent à Amazon DocumentDB via une chaîne de connexion ne rencontreront pas ce problème, car cette valeur `ClusterConnectionMode` sera définie `MULTIPLE` lors de la définition des préférences de lecture.

```
MongoClientURI mongoClientURI = new MongoClientURI("mongodb://usr:pass:cluster-
endpoint:27317/test?ssl=false&replicaSet=rs0&readpreference=secondaryPreferred");
MongoClient mongoClient = MongoClients.create(mongoClientURI.getURI());
```

  Ou en utilisant `MongoClientSettings` Builder avec la `applyConnectionString` méthode.

```
final MongoClientSettings settings = MongoClientSettings.builder()
        .credential(credential)
        .applyConnectionString(new ConnectionString("usr:pass:cluster-endpoint:27317/
test?ssl=false&replicaSet=rs0&readpreference=secondaryPreferred"))
        .retryWrites(false)
        .applyToSslSettings(builder # builder
                .enabled(false))
        .build();
MongoClient mongoClient = MongoClients.create(settings);
```

- Paramétré explicitement `ClusterConnectionMode` sur`MULTIPLE`. Cela n'est nécessaire que lors de l'utilisation `applyToClusterSettings` de e`thosts().size() == 1`.

```
final ServerAddress serverAddress0 = new ServerAddress("cluster-endpoint", 27317));
final MongoCredential credential = MongoCredential.createCredential("xxx","admin",
 "xxxx".toCharArray());
final MongoClientSettings settings = MongoClientSettings.builder()
    .credential(credential)
    .readPreference(ReadPreference.secondaryPreferred())
    .retryWrites(false)
    .applyToSslSettings(builder # builder
    .enabled(false))
    .applyToClusterSettings(builder # builder
                .hosts(Arrays.asList(serverAddress0))
```

```
                .requiredReplicaSetName("rs0"))
                .mode(ClusterConnectionMode.MULTIPLE))
    .build();
MongoClient mongoClient = MongoClients.create(settings);
```

Test d'une connexion à une instance Amazon DocumentDB

Vous pouvez tester votre connexion à un cluster à l'aide des outils courants Linux ou Windows.

Depuis un terminal Linux ou Unix, testez la connexion en saisissant les informations suivantes (remplacez `cluster-endpoint` par le point de terminaison et `port` par le port de votre instance) :

```
nc -zv cluster-endpoint port
```

Voici un exemple d'opération et la valeur de retour :

```
nc -zv docdbTest.d4c7nm7stsfc0.us-west-2.docdb.amazonaws.com 27017

Connection to docdbTest.d4c7nm7stsfc0.us-west-2.docdb.amazonaws.com 27017 port [tcp/*]
 succeeded!
```

Connexion à un point de terminaison non valide

Lorsque vous vous connectez à un cluster Amazon DocumentDB et que vous utilisez un point de terminaison de cluster non valide, une erreur similaire à la suivante apparaît.

```
mongo --ssl \
   --host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 \
   --sslCAFile global-bundle.pem \
   --username <user-name> \
   --password <password>
```

Le résultat se présente comme suit :

```
MongoDB shell version v3.6
connecting to: mongodb://sample-cluster.node.us-east-1.docdb.amazonaws.com:27017/
2018-11-14T17:21:18.516-0800 I NETWORK [thread1] getaddrinfo("sample-cluster.node.us-
east-1.docdb.amazonaws.com") failed:
nodename nor servname provided, or not known 2018-11-14T17:21:18.537-0800 E QUERY
 [thread1] Error: couldn't initialize
```

```
connection to host sample-cluster.node.us-east-1.docdb.amazonaws.com, address is
 invalid :
connect@src/mongo/shell/mongo.js:237:13@(connect):1:6
exception: connect failed
```

Pour connaître le point de terminaison valide pour un cluster, exécutez la commande suivante :

```
aws docdb describe-db-clusters \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].[Endpoint,Port]'
```

Pour connaître le point de terminaison valide pour une instance, exécutez la commande suivante :

```
aws docdb describe-db-instances \
   --db-instance-identifier sample-instance \
   --query 'DBInstances[*].[Endpoint.Address,Endpoint.Port]'
```

Pour plus d'informations, consultez Comprendre les points de terminaison Amazon DocumentDB.

La configuration du pilote a un impact sur le nombre de connexions

Lorsque vous utilisez le pilote client pour vous connecter à un cluster Amazon DocumentDB, il est important de prendre en compte le paramètre de `maxPoolSize` configuration. Le `maxPoolSize` paramètre détermine le nombre maximal de connexions que le pilote client conservera dans son pool de connexions.

Création d'index

Les rubriques suivantes expliquent la procédure à suivre en cas d'échec de la génération de votre index ou de votre index d'arrière-plan.

Rubriques

- La création de l'index échoue
- Problèmes et échecs de latence lors de la création de l'index d'arrière-plan

La création de l'index échoue

Amazon DocumentDB utilise le stockage local sur une instance dans le cadre du processus de création d'index. Vous pouvez surveiller cette utilisation du disque à l'aide de la CloudWatch métrique

FreeLocalde stockage (`CloudWatch -> Metrics -> DocDB -> Instance Metrics`). Lorsque la génération d'un index utilise l'ensemble du disque local et échoue, vous recevez une erreur. Lorsque vous migrez des données vers Amazon DocumentDB, nous vous encourageons à créer d'abord des index, puis à insérer les données. Pour plus d'informations sur les stratégies de migration et la création d'index, consultez Migration vers Amazon DocumentDB la documentation Amazon DocumentDB et le blog : Migrer de MongoDB vers Amazon DocumentDB en utilisant la méthode hors ligne.

Lorsque vous créez des index sur un cluster existant, si la création de l'index prend plus de temps que prévu ou échoue, nous vous recommandons de redimensionner l'instance pour créer l'index, puis de la réduire une fois l'index créé. Amazon DocumentDB vous permet de redimensionner rapidement la taille des instances en quelques minutes à l'aide du AWS Management Console ou du. AWS CLI Pour plus d'informations, consultez Gestion de classes d'instance. Avec la tarification des instances à la seconde, vous payez seulement pour les ressources que vous utilisez à la seconde près.

Problèmes et échecs de latence lors de la création de l'index d'arrière-plan

Les compilations d'index en arrière-plan dans Amazon DocumentDB ne démarrent que lorsque toutes les requêtes sur l'instance principale lancées avant le lancement de la génération d'index ne sont terminées. Si la requête est longue, les compilations d'index en arrière-plan seront bloquées jusqu'à la fin de la requête et peuvent donc prendre plus de temps que prévu. Cela est vrai même si les collections sont vides.

Les constructions d'index de premier plan ne présentent pas le même comportement de blocage. Au lieu de cela, les constructions d'index de premier plan bloquent exclusivement la collection jusqu'à ce que la création de l'index soit terminée. Ainsi, pour créer des index sur une collection vide et éviter de bloquer les requêtes de longue durée, nous vous suggérons d'utiliser des versions d'index de premier plan.

> **Note**
>
> Amazon DocumentDB autorise la création d'un seul index d'arrière-plan sur une collection à la fois. Si les opérations DDL (langage de définition de données) telles que `createIndex()` ou `dropIndex()` se produisent pendant la génération d'un index d'arrière-plan, celle-ci échoue.

Performances et utilisation des ressources

Cette section fournit des questions et des solutions aux problèmes de diagnostic courants liés aux déploiements Amazon DocumentDB. Les exemples fournis utilisent le shell mongo et sont limités à une instance individuelle. Pour rechercher un point de terminaison d'instance, consultez Comprendre les points de terminaison Amazon DocumentDB.

Rubriques

Comment puis-je déterminer le nombre d'opérations d'insertion, de mise à jour et de suppression effectuées sur ma collection via l'API Mongo ?

Pour afficher le nombre d'opérations d'insertion, de mise à jour et de suppression effectuées sur une collection donnée, exécutez la commande suivante sur cette collection :

```
db.collection.stats()
```

Le résultat de cette commande décrit ce qui suit dans son `opCounters` champ :

- numDocsIns- Le nombre de documents insérés dans cette collection. Cela inclut les documents insérés à l'aide `insert` des `insertMany` commandes et, ainsi que les documents insérés par un upsert.
- numDocsUpd- Le nombre de documents mis à jour dans cette collection. Cela inclut les documents mis à jour à l'aide `update` des `findAndModify` commandes et.
- numDocsDel- Le nombre de documents supprimés de cette collection. Cela inclut les documents supprimés à l'aide des `findAndModify` commandes `deleteOne` `deleteManyremove`,, et.
- LastReset : heure à laquelle ces compteurs ont été réinitialisés pour la dernière fois. Les statistiques fournies par cette commande sont réinitialisées lors du démarrage/arrêt du cluster ou lors de la redimensionnement/réduction de l'instance.

Un exemple de résultat de l'exécution `db.collection.stats()` est présenté ci-dessous.

```
{
    "ns" : "db.test",
    "count" : ...,
    "size" : ...,
    "avgObjSize" : ...,
    "storageSize" : ...,
    "capped" : false,
    "nindexes" : ...,
    "totalIndexSize" : ...,
    "indexSizes" : {
        "_id_" : ...,
        "x_1" : ...
    },
    "collScans" : ...,
    "idxScans" : ...,
    "opCounter" : {
        "numDocsIns" : ...,
        "numDocsUpd" : ...,
        "numDocsDel" : ...
    },
    "cacheStats" : {
        "collBlksHit" : ...,
        "collBlksRead" : ..,
        "collHitRatio" : ...,
        "idxBlksHit" : ...,
```

```
        "idxBlksRead" : ...,
        "idxHitRatio" : ...
    },
    "lastReset" : "2022-09-02 19:41:40.471473+00",
    "ok" : 1,
    "operationTime" : Timestamp(1662159707, 1)
}
```

Cette commande de statistiques doit être utilisée lors de l'affichage des compteurs spécifiques à une collection pour les opérations d'insertion, de mise à jour et de suppression via l'API Mongo. Vous pouvez également consulter les compteurs d'opérations spécifiques à une collection en activant l'audit DML. Le nombre d'opérations d'insertion, de mise à jour et de suppression effectuées sur toutes les collections pendant des intervalles d'une minute peut être visualisé dansSurveillance d'Amazon DocumentDB avec CloudWatch.

Comment analyser les performances du cache ?

L'analyse des performances du cache peut donner un aperçu de l'efficacité de la récupération des données et des performances du système. Elle est basée sur la quantité de données lues sur le disque par rapport au cache. Nous fournissons des statistiques sur le cache concernant le nombre d'accès au cache (données lues dans le cache) et de défaillances du cache (données introuvables dans le cache et lues sur le disque) afin de donner un aperçu des performances du cache. Les statistiques de cache d'une collection spécifique peuvent être trouvées en exécutant la commande suivante sur cette collection :

```
db.collection.stats()
```

Les valeurs du `cacheStats` champ en sortie de cette commande fournissent des statistiques de cache pour la collection ainsi que les statistiques de cache totales pour les index créés sur la collection. Ces statistiques sont répertoriées ci-dessous :

- **`collBlksHit`**- Le nombre de blocs lus depuis le cache lors des opérations sur cette collection.
- **`collBlksRead`**- Le nombre de blocs lus sur le disque (cache manquant) lors des opérations sur cette collection.
- **`collHitRatio`**- Le taux de réussite du cache pour cette collection (`100 * [collBlksHit / (collBlksHit + collBlksRead)]`).
- **`idxBlksHit`**- Le nombre de blocs lus depuis le cache pour tout index créé sur cette collection.

- **idxBlksRead**- Le nombre de blocs lus sur le disque (cache manquant) pour tout index créé sur cette collection.
- **idxHitRatio**- Le taux de réussite du cache pour les index créés dans cette collection (`100 * [idxBlksHit / (idxBlksHit + idxBlksRead)]`).
- **lastReset**- Heure à laquelle ces statistiques ont été réinitialisées pour la dernière fois. Les statistiques fournies par `db.collection.stats()` sont réinitialisées lors du démarrage/arrêt du cluster ou lors de la redimensionnement/réduction de l'instance.

Une ventilation des `idxBlksRead` champs `idxBlksHit` et pour chaque index peut également être trouvée à l'aide de la `indexStats` commande. Les statistiques de cache spécifiques à l'index peuvent être trouvées en exécutant la commande suivante :

```
db.collection.aggregate([{$indexStats:{}}]).pretty()
```

Pour chaque index, les statistiques de cache suivantes se trouvent sous le `cacheStats` champ :

- **blksHit**- Le nombre de blocs lus depuis le cache pour cet index.
- **blksRead**- Le nombre de blocs lus sur le disque pour cet index.
- **blksHitRatio**- Le taux de réussite du cache arrondi à quatre décimales, calculé par`100 * [blksHit / (blksHit + blksRead)]`.

Comment rechercher les requêtes de longue durée ou bloquées et les arrêter ?

Les requêtes utilisateur peuvent s'exécuter lentement en raison d'un plan de requête sous-optimal ou peuvent être bloquées en raison d'un conflit de ressources.

Pour rechercher les requêtes de longue durée qui ralentissent en raison d'un plan de requête sous-optimal ou les requêtes bloquées en raison d'un conflit de ressources, utilisez la commande `currentOp`. Vous pouvez filtrer la commande afin de réduire la liste des requêtes pertinentes à résilier. Vous devez avoir `opid` associé à la requête de longue durée pour pouvoir mettre fin à la requête.

La requête suivante utilise la commande `currentOp` pour répertorier toutes les requêtes qui sont bloquées ou en cours d'exécution pendant plus de 10 secondes.

```
db.adminCommand({
    aggregate: 1,
    pipeline: [
        {$currentOp: {}},
        {$match:
            {$or: [
                {secs_running: {$gt: 10}},
                {WaitState: {$exists: true}}]}},
        {$project: {_id:0, opid: 1, secs_running: 1}}],
    cursor: {}
});
```

Ensuite, vous pouvez affiner la requête pour trouver le `opid` d'une requête en cours d'exécution pendant plus de 10 secondes et y mettre fin.

Pour rechercher et résilier une requête exécutée depuis plus de 10 secondes

1. Recherchez l'`opid` de la requête.

   ```
   db.adminCommand({
       aggregate: 1,
       pipeline: [
           {$currentOp: {}},
           {$match:
               {$or:
                   [{secs_running: {$gt: 10}},
                    {WaitState: {$exists: true}}]}}],
       cursor: {}
   });
   ```

   La sortie de cette opération ressemble à ceci (format JSON).

   ```
   {
       "waitedMS" : NumberLong(0),
       "cursor" : {
           "firstBatch" : [
               {
                   "opid" : 24646,
                   "secs_running" : 12
               }
           ],
           "id" : NumberLong(0),
   ```

```
        "ns" : "admin.$cmd"
    },
    "ok" : 1
}
```

2. Résiliez la requête à l'aide de l'opération `killOp`.

```
db.adminCommand({killOp: 1, op: 24646});
```

Comment puis-je consulter un plan de requête et optimiser une requête ?

Si une requête s'exécute lentement, cela peut être dû au fait que l'exécution de la requête nécessite une analyse complète de la collection pour choisir les documents pertinents. Parfois, la création d'index appropriés permet à la requête de s'exécuter plus rapidement. Pour détecter ce scénario et choisir les champs sur lesquels créer les index, utilisez la commande `explain`.

> **Note**
>
> Amazon DocumentDB émule l'API MongoDB 3.6 sur un moteur de base de données spécialement conçu qui utilise un système de stockage distribué, tolérant aux pannes et autoréparateur. Par conséquent, les plans de requête et le résultat de `explain()` peuvent différer entre Amazon DocumentDB et MongoDB. Les clients qui souhaitent contrôler leur plan de requête peuvent utiliser l'opérateur `$hint` pour appliquer la sélection d'un index préféré.

Exécutez la requête que vous souhaitez améliorer sous la commande `explain` comme suit.

```
db.runCommand({explain: {<query document>}})
```

Voici un exemple d'opération.

```
db.runCommand({explain:{
    aggregate: "sample-document",
    pipeline: [{$match: {x: {$eq: 1}}}],
    cursor: {batchSize: 1}}
});
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "db.test",
        "winningPlan" : {
            "stage" : "COLLSCAN"
        }
    },
    "serverInfo" : {
        "host" : "...",
        "port" : ...,
        "version" : "..."
    },
    "ok" : 1
}
```

La sortie ci-dessus indique que l'étape `$match` exige d'analyser l'ensemble des collections et de vérifier si le champ `"x"` de chaque document est égal à 1. Si la collection comporte trop de documents, l'analyse de la collection et, donc, les performances globales de la requête seront très lentes. Ainsi, la présence de `"COLLSCAN"` dans la sortie de la commande `explain` indique que les performances de requête peuvent être améliorées en créant des index appropriés.

Dans cet exemple, la requête vérifie si le champ `"x"` est égal à 1 dans tous les documents. Par conséquent, la création d'un index sur le champ `"x"` permet à la requête d'éviter l'analyse complète de la collection et d'utiliser l'index pour renvoyer les documents pertinents plus rapidement.

Après avoir créé un index sur le champ `"x"`, la sortie d'`explain` se présente comme suit.

```
{
    "queryPlanner" : {
         "plannerVersion" : 1,
         "namespace" : "db.test",
         "winningPlan" : {
             "stage" : "IXSCAN",
             "indexName" : "x_1",
             "direction" : "forward"
         }
    },
    "serverInfo" : {
        "host" : "...",
        "port" : ...,
        "version" : "..."
```

```
    },
    "ok" : 1
}
```

Ainsi, la création d'un index sur le champ `"x"` a permis à l'étape `$match` d'utiliser une analyse d'index pour réduire le nombre de documents sur lesquels le prédicat `"x = 1"` doit être évalué.

Pour les petites collections, le processeur de requêtes Amazon DocumentDB peut choisir de ne pas utiliser d'index si les gains de performances sont négligeables.

Comment puis-je voir un plan de requête dans des clusters élastiques ?

Pour examiner un plan de requête dans des clusters élastiques, utilisez la `explain` commande. Voici un exemple d'`explain`opération sur une requête de recherche ciblant une collection fragmentée :

```
db.runCommand(
   {
     explain: { find: "cities", filter: {"name": "Seoul"}}
   }
)
```

> **Note**
>
> Amazon DocumentDB émule MongoDB sur un moteur de base de données spécialement conçu. Par conséquent, les plans de requête et le résultat de `explain()` peuvent différer entre Amazon DocumentDB et MongoDB. Vous pouvez contrôler le plan de requête à l'aide de l'`$hint`opérateur pour imposer la sélection d'un index préféré.

Le résultat de cette opération peut ressembler à ce qui suit (format JSON) :

```
{
  "queryPlanner" : {
    "elasticPlannerVersion" : 1,
    "winningPlan" : {
      "stage" : "SINGLE_SHARD",
      "shards" : [
        {
          "plannerVersion" : 1,
          "namespace" : "population.cities",
```

```
          "winningPlan" : {
            "stage" : "SHARD_MERGE",
            "shards" : [
              {
                "shardName" : "f2cf5cfd-fe9c-40ca-b4e5-298ca0d11111",
                "plannerVersion" : 1,
                "namespace" : "population.cities",
                "winningPlan" : {
                  "stage" : "PARTITION_MERGE",
                  "inputStages" : [
                    {
                      "stage" : "COLLSCAN",
                      "partitionCount" : 21
                    }
                  ]
                }
              },
              {
                "shardName" : "8f3f80e2-f96c-446e-8e9d-aab8c7f22222",
                "plannerVersion" : 1,
                "namespace" : "population.cities",
                "winningPlan" : {
                  "stage" : "PARTITION_MERGE",
                  "inputStages" : [
                    {
                      "stage" : "COLLSCAN",
                      "partitionCount" : 21
                    }
                  ]
                }
              },
              {
                "shardName" : "32c5a06f-1b2b-4af1-8849-d7c4a033333",
                "plannerVersion" : 1,
                "namespace" : "population.cities",
                "winningPlan" : {
                  "stage" : "PARTITION_MERGE",
                  "inputStages" : [
                    {
                      "stage" : "COLLSCAN",
                      "partitionCount" : 22
                    }
                  ]
                }
```

```
              }
            ]
          },
          "shardName" : "32c5a06f-1b2b-4af1-8849-d7c4a0f3fb58"
        }
      ]
    }
  },
  "serverInfo" : {
    "host" : "example-4788267630.us-east-1.docdb-elastic.amazonaws.com:27017",
    "version" : "5.0.0"
  },
  "ok" : 1,
  "operationTime" : Timestamp(1695097923, 1)
}
```

La sortie précédente montre le plan de requête pour la `find` requête sur un cluster à trois partitions. Chaque partition possède plusieurs partitions de données qui peuvent avoir différents étages d'entrée. Dans cet exemple, un « COLLSCAN » (une analyse de collection) est exécuté sur toutes les partitions avant que les résultats ne soient fusionnés à l'étape « PARTITION_MERGE » au sein de chaque partition. Les résultats des partitions sont ensuite fusionnés à l'étape « SHARD_MERGE » avant d'être renvoyés au client.

Quelle est la marche à suivre pour répertorier toutes les opérations en cours d'exécution sur une instance ?

En tant qu'utilisateur ou utilisateur principal, vous souhaitez souvent répertorier toutes les opérations en cours d'exécution sur une instance à des fins de diagnostic et de dépannage. (Pour de plus amples informations sur la gestion des utilisateurs, veuillez consulter Gestion des utilisateurs Amazon DocumentDB.)

Avec le `mongo` shell, vous pouvez utiliser la requête suivante pour répertorier toutes les opérations en cours sur une instance Amazon DocumentDB.

```
db.adminCommand({currentOp: 1, $all: 1});
```

La requête renvoie la liste complète de toutes les demandes d'utilisateur et des tâches système internes en cours d'exécution sur l'instance.

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "inprog" : [
        {
            "desc" : "INTERNAL"
        },
        {
            "desc" : "TTLMonitor",
            "active" : false
        },
        {
            "client" : ...,
            "desc" : "Conn",
            "active" : true,
            "killPending" : false,
            "opid" : 195,
            "ns" : "admin.$cmd",
            "command" : {
                "currentOp" : 1,
                "$all" : 1
            },
            "op" : "command",
            "$db" : "admin",
            "secs_running" : 0,
            "microsecs_running" : NumberLong(68),
            "clientMetaData" : {
            "application" : {
                "name" : "MongoDB Shell"
            },
            "driver" : {
                ...
            },
            "os" : {
                ...
            }
          }
       },
       {
          "desc": "GARBAGE_COLLECTION",
          "garbageCollection": {
             "databaseName": "testdb",
             "collectionName": "testCollectionA"
          },
          "secs_running": 3,
```

```
            "microsecs_running": NumberLong(3123456)
         },
         {
            "desc": "GARBAGE_COLLECTION",
            "garbageCollection": {
               "databaseName": "testdb",
               "collectionName": "testCollectionB"
            },
            "secs_running": 4,
            "microsecs_running": NumberLong(4123456)
         }
      ],
      "ok" : 1
}
```

Les valeurs suivantes sont valides pour le champ `"desc"` :

- **`INTERNAL`**— Tâches internes au système, telles que le nettoyage du curseur ou les tâches de nettoyage des utilisateurs obsolètes.
- **`TTLMonitor`**— Le fil de surveillance Time to Live (TTL). Son statut d'exécution est reflété dans le champ `"active"`.
- **`GARBAGE_COLLECTION`**— Le fil interne du ramasse-miettes.
- **`CONN`**— La requête de l'utilisateur.
- **`CURSOR`**— L'opération est un curseur inactif qui attend que l'utilisateur appelle la commande « GetMore » pour obtenir le prochain lot de résultats. Dans cet état, le curseur consomme de la mémoire, mais ne consomme aucun calcul.

La sortie précédente répertorie également toutes les requêtes utilisateur en cours d'exécution dans le système. Chaque requête utilisateur s'exécute dans le contexte d'une base de données et d'une collection, et l'union de ces deux éléments représente un espace de noms. L'espace de noms de chaque requête utilisateur est disponible dans le champ `"ns"`.

Parfois, vous devez répertorier toutes les requêtes utilisateur qui sont en cours d'exécution dans un espace de noms particulier. Par conséquent, la sortie précédente doit être filtrée sur le champ `"ns"`. Voici un exemple de requête pour obtenir la sortie à filtrer. La requête répertorie toutes les requêtes utilisateur en cours d'exécution dans la base de données `"db"` et la collection `"test"` (c'est-à-dire, l'espace de noms `"db.test"`).

```
db.adminCommand({aggregate: 1,
```

```
    pipeline: [{$currentOp: {allUsers: true, idleConnections: true}},
               {$match: {ns: {$eq: "db.test"}}}],
    cursor: {}
});
```

En tant qu'utilisateur principal du système, vous pouvez voir les requêtes de tous les utilisateurs ainsi que toutes les tâches internes du système. Tous les autres utilisateurs peuvent uniquement voir leurs propres requêtes.

Si le nombre total de requêtes et de tâches système internes dépasse la taille du curseur de traitement par lots par défaut, le shell mongo génère automatiquement un objet itérateur `'it'` pour afficher le reste des résultats. Il continue d'exécuter la commande `'it'` jusqu'à ce que tous les résultats aient été affichés.

Comment savoir si une requête progresse ?

Les requêtes utilisateurs peuvent s'exécuter lentement en raison d'un plan de requête mal adapté ou être bloquées en raison d'un conflit de ressource. Le débogage de telles requêtes est un processus en plusieurs étapes qui nécessite parfois d'exécuter la même étape plusieurs fois.

La première étape de débogage consiste à répertorier toutes les requêtes de longue durée ou bloquées. La requête suivante répertorie toutes les requêtes utilisateur qui ont été exécutées depuis plus de 10 secondes ou qui attendent des ressources.

```
db.adminCommand({aggregate: 1,
                 pipeline: [{$currentOp: {}},
                            {$match: {$or: [{secs_running: {$gt: 10}},
                                            {WaitState: {$exists: true}}]}},
                            {$project: {_id:0,
                                        opid: 1,
                                        secs_running: 1,
                                        WaitState: 1,
                                        blockedOn: 1,
                                        command: 1}}],
                 cursor: {}
                });
```

Répétez régulièrement la requête précédente pour déterminer si la liste des requêtes change et pour identifier les requêtes de longue durée ou bloquées.

Si le document de sortie pour la requête pertinente a un champ `WaitState`, cela indique que la requête s'exécute lentement ou est bloquée à cause d'un conflit de ressource. Le conflit de ressource peut être dû à des E/S, des tâches système internes ou d'autres requêtes utilisateur.

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "waitedMS" : NumberLong(0),
    "cursor" : {
        "firstBatch" : [
            {
                "opid" : 201,
                "command" : {
                    "aggregate" : ...
                },
                "secs_running" : 208,
                "WaitState" : "IO"
            }
        ],
        "id" : NumberLong(0),
        "ns" : "admin.$cmd"
    },
    "ok" : 1
}
```

Les E/S peuvent constituer un goulot d'étranglement si un trop grand nombre de requêtes s'exécutent simultanément dans différentes collections sur la même instance, ou si la taille d'instance est trop petite pour l'ensemble de données sur lequel la requête s'exécute. Si les requêtes sont en lecture seule, vous pouvez atténuer la première situation en séparant les requêtes de chaque collection entre des réplicas distincts. Pour les mises à jour simultanées dans différentes collections ou quand la taille d'instance est trop petite pour l'ensemble de données, la solution consiste à faire monter l'instance en puissance.

Si le conflit de ressource est dû à d'autres requêtes utilisateur, le champ `"blockedOn"` dans le document de sortie comporte la valeur `"opid"` de la requête qui affecte cette requête. À l'aide de la valeur `"opid"`, suivez la chaîne des champs `"WaitState"` et `"blockedOn"` de toutes les requêtes pour trouver la requête en tête de la chaîne.

Si la tâche en tête de la chaîne est une tâche interne, la seule solution consiste à arrêter la requête et à la réexécuter après un certain temps.

Voici un exemple de sortie dans lequel la requête de recherche est bloquée sur un verrou de collection détenu par une autre tâche.

```
{
    "inprog" : [
        {
            "client" : "...",
            "desc" : "Conn",
            "active" : true,
            "killPending" : false,
            "opid" : 75,
            "ns" : "...",
            "command" : {
                "find" : "...",
                "filter" : {

                }
            },
            "op" : "query",
            "$db" : "test",
            "secs_running" : 9,
            "microsecs_running" : NumberLong(9449440),
            "threadId" : 24773,
            "clientMetaData" : {
                "application" : {
                   "name" : "MongoDB Shell"
                },
                "driver" : {
                    ...
                },
                "os" : {
                    ...
                }
            },
            "WaitState" : "CollectionLock",
            "blockedOn" : "INTERNAL"
        },
        {
            "desc" : "INTERNAL"
        },
        {
            "client" : "...",
            ...
```

```
            "command" : {
                "currentOp" : 1
            },
            ...
        }
    ],
    "ok" : 1
}
```

Si `"WaitState"` a les valeurs `"Latch"`, `"SystemLock"`, `"BufferLock"`, `"BackgroundActivity"` ou `"Other"`, des tâches système internes sont à l'origine du conflit de ressource. Si la situation persiste pendant longtemps, la seule atténuation consiste à mettre fin à la requête et à l'exécuter à nouveau ultérieurement.

Comment puis-je déterminer pourquoi un système fonctionne soudainement lentement ?

Voici quelques raisons courantes du ralentissement d'un système :

- Contention de ressources excessives entre les requêtes simultanées
- Nombre de requêtes simultanées actives augmentant au fil du temps
- Tâches système internes telles que `"GARBAGE_COLLECTION"`

Pour surveiller l'utilisation du système au fil du temps, exécutez la requête `"currentOp"` suivante périodiquement et placez les résultats dans un magasin externe. La requête compte le nombre de requêtes et d'opérations dans chaque espace de noms au sein du système. Les résultats de l'utilisation du système peuvent être ensuite analysés pour comprendre la charge du système et prendre les décisions appropriées.

```
db.adminCommand({aggregate: 1,
                 pipeline: [{$currentOp: {allUsers: true, idleConnections: true}},
                            {$group: {_id: {desc: "$desc", ns: "$ns", WaitState:
 "$WaitState"}, count: {$sum: 1}}}],
                 cursor: {}
                });
```

Cette requête renvoie un agrégat de toutes les requêtes en cours d'exécution dans chaque espace de noms et de toutes les tâches système internes, ainsi que le nombre d'états d'attente par espace de noms, le cas échéant.

La sortie de cette opération ressemble à ceci (format JSON).

```
{
    "waitedMS" : NumberLong(0),
    "cursor" : {
        "firstBatch" : [
            {
                "_id" : {
                    "desc" : "Conn",
                    "ns" : "db.test",
                    "WaitState" : "CollectionLock"
                },
               "count" : 2
            },
            {
                "_id" : {
                    "desc" : "Conn",
                    "ns" : "admin.$cmd"
                },
                "count" : 1
            },
            {
                "_id" : {
                    "desc" : "TTLMonitor"
                },
                "count" : 1
            }
        ],
        "id" : NumberLong(0),
        "ns" : "admin.$cmd"
    },
    "ok" : 1
}
```

Dans la sortie précédente, il existe 2 requêtes utilisateur dans l'espace de noms `"db.test"` qui sont bloquées sur un verrouillage de collecte : 1 requête dans l'espace de noms `"admin.$cmd"` et une tâche interne `"TTLMonitor"`.

Si la sortie indique de nombreuses requêtes avec des états d'attente bloquants, consultez Comment rechercher les requêtes de longue durée ou bloquées et les arrêter ?

Comment déterminer la cause d'une utilisation élevée du processeur sur une ou plusieurs instances de cluster ?

Les sections suivantes peuvent vous aider à identifier la cause d'une utilisation élevée de l'UC des instances. Vos résultats peuvent varier en fonction de la charge de travail.

- Pour déterminer pourquoi une instance s'exécute lentement soudainement, consultez Comment puis-je déterminer pourquoi un système fonctionne soudainement lentement ?
- Pour identifier et mettre fin aux requêtes de longue durée sur une instance particulière, consultez Comment rechercher les requêtes de longue durée ou bloquées et les arrêter ?
- Pour comprendre si une requête progresse, consultez Comment savoir si une requête progresse ?
- Pour déterminer pourquoi l'exécution d'une requête prend beaucoup de temps, consultez Comment puis-je consulter un plan de requête et optimiser une requête ?
- Pour suivre les requêtes de longue durée au fil du temps, consultez Profilage des opérations Amazon DocumentDB.

En fonction de la raison de votre utilisation élevée de l'UC de l'instance, l'exécution d'une ou de plusieurs des actions suivantes peut vous aider.

- Si l'instance principale présente une utilisation élevée de l'UC, ce qui n'est pas le cas des instances de réplica, envisagez de répartir le trafic en lecture entre les réplicas via les paramètres de préférence en lecture du client (par exemple, `secondaryPreferred`). Pour plus d'informations, consultez Connexion à Amazon DocumentDB en tant qu'ensemble de réplicas.

  L'utilisation de réplicas pour les lectures peut mieux utiliser les ressources du cluster en permettant à l'instance principale de traiter plus de trafic d'écriture. Les lectures de réplicas sont cohérentes à terme.
- Si l'utilisation élevée de l'UC est le résultat de votre charge de travail en écriture, le fait de remplacer la taille des instances du cluster par un type d'instance plus grand augmente le nombre de cœurs d'UC disponibles pour traiter la charge de travail. Pour plus d'informations, consultez instances et Spécifications de la classe d'instance.
- Si toutes les instances de cluster présentent une utilisation élevée de l'UC et que la charge de travail utilise des réplicas pour les lectures, l'ajout d'autres réplicas au cluster augmente les ressources disponibles pour le trafic en lecture. Pour plus d'informations, consultez Ajouter une instance Amazon DocumentDB à un cluster.

Comment déterminer les curseurs ouverts sur une instance ?

Lorsque vous êtes connecté à une instance Amazon DocumentDB, vous pouvez utiliser la commande `db.runCommand("listCursors")` pour répertorier les curseurs ouverts sur cette instance. Il existe une limite de 4 560 curseurs actifs ouverts à tout moment sur une instance Amazon DocumentDB donnée, en fonction du type d'instance. Il est généralement conseillé de fermer les curseurs qui ne sont plus utilisés car les curseurs utilisent des ressources sur une instance et ont une limite supérieure. Consultez Quotas et limites Amazon DocumentDB les limites spécifiques.

```
db.runCommand("listCursors")
```

Comment puis-je déterminer la version actuelle du moteur Amazon DocumentDB ?

Pour déterminer la version actuelle de votre moteur Amazon DocumentDB, exécutez la commande suivante.

```
db.runCommand({getEngineVersion: 1})
```

La sortie de cette opération ressemble à ceci (format JSON).

```
{ "engineVersion" : "2.x.x", "ok" : 1 }
```

> **Note**
>
> La version du moteur pour Amazon DocumentDB 3.6 est 1.x.x et la version du moteur pour Amazon DocumentDB 4.0 est 2.x.x.

Comment analyser l'utilisation des index et identifier les index inutilisés ?

Pour identifier les index d'une collection donnée, exécutez la commande suivante :

```
db.collection.getIndexes()
```

Pour analyser la quantité d'index utilisés lors des opérations effectuées sur les collections, les `indexStats` commandes `collStats` et peuvent être utilisées. Pour afficher le nombre total de

scans effectués à l'aide d'index (scans d'index) par rapport au nombre de scans effectués sans index (scans de collection), exécutez la commande suivante :

```
db.collection.stats()
```

La sortie de cette commande inclut les valeurs suivantes :

- **`idxScans`**- Le nombre de scans effectués sur cette collection à l'aide d'un index.
- **`collScans`**- Le nombre de scans effectués sur cette collection sans utiliser d'index. Ces scans auraient impliqué l'examen des documents de la collection un par un.
- **`lastReset`**- L'heure à laquelle ces compteurs ont été réinitialisés pour la dernière fois. Les statistiques fournies par cette commande sont réinitialisées lors du démarrage/arrêt du cluster ou lors de la redimensionnement/réduction de l'instance.

Le détail de l'utilisation de chaque index se trouve dans le résultat de la commande suivante. Il est recommandé d'identifier et de supprimer régulièrement les index inutilisés afin d'améliorer les performances et de réduire les coûts, car cela élimine le calcul, le stockage et les E/S inutiles utilisés pour maintenir les index.

```
db.collection.aggregate([{$indexStats:{}}]).pretty()
```

Le résultat de cette commande donne les valeurs suivantes pour chaque index créé dans la collection :

- **`ops`**- Le nombre d'opérations ayant utilisé l'index. Si votre charge de travail est en cours d'exécution depuis suffisamment longtemps et que vous êtes sûr qu'elle est stable, une valeur `ops` de zéro indique que l'indice n'est pas utilisé du tout.
- **`numDocsRead`**- Le nombre de documents lus lors des opérations utilisant cet indice.
- **`since`**- Le temps écoulé depuis qu'Amazon DocumentDB a commencé à collecter des statistiques sur l'utilisation des index, qui correspond généralement à la valeur écoulée depuis le dernier redémarrage de la base de données ou la dernière action de maintenance.
- **`size`**- La taille de cet index en octets.

L'exemple suivant est un exemple de résultat de l'exécution de la commande ci-dessus :

```
{
```

```
    "name" : "_id_",
    "key" : {
        "_id" : 1
    },
    "host" : "example-host.com:12345",
    "size" : NumberLong(...),
    "accesses" : {
        "ops" : NumberLong(...),
        "docsRead" : NumberLong(...),
        "since" : ISODate("...")
    },
    "cacheStats" : {
        "blksRead" : NumberLong(...),
        "blksHit" : NumberLong(...),
        "hitRatio" : ...
    }
}
{
    "name" : "x_1",
    "key" : {
        "x" : 1
    },
    "host" : "example-host.com:12345",
    "size" : NumberLong(...),
    "accesses" : {
        "ops" : NumberLong(...),
        "docsRead" : NumberLong(...),
        "since" : ISODate("...")
    },
    "cacheStats" : {
        "blksRead" : NumberLong(...),
        "blksHit" : NumberLong(...),
        "hitRatio" : ...
    }
}
```

Pour déterminer la taille d'index globale d'une collection, exécutez la commande suivante :

```
db.collection.stats()
```

Pour supprimer un index non utilisé, exécutez la commande suivante :

```
db.collection.dropIndex("indexName")
```

Comment identifier les index manquants ?

Vous pouvez utiliser le profileur Amazon DocumentDB pour enregistrer les requêtes lentes. Une requête qui apparaît à plusieurs reprises dans le journal des requêtes lentes peut indiquer qu'un index supplémentaire est nécessaire pour améliorer les performances de cette requête.

Vous pouvez identifier les possibilités d'index utiles en recherchant les requêtes longues dont une ou plusieurs étapes exécutent au moins une étape `COLLSCAN`, qui signifie que l'étape de requête doit lire chaque document de la collection afin de fournir une réponse à la requête.

L'exemple suivant montre une requête sur une collection de trajets de taxi exécutée sur une collection de taille importante.

```
db.rides.count({"fare.totalAmount":{$gt:10.0}}))
```

Pour exécuter cet exemple, la requête devait effectuer une analyse de collection (c'est-à-dire lire chaque document de la collection) car il n'y a pas d'index sur le champ `fare.totalAmount`. Le résultat du profileur Amazon DocumentDB pour cette requête ressemble à ce qui suit :

```
{
    ...
    "cursorExhausted": true,
    "nreturned": 0,
    "responseLength": 0,
    "protocol": "op_query",
    "millis": 300679,
    "planSummary": "COLLSCAN",
    "execStats": {
        "stage": "COLLSCAN",
        "nReturned": "0",
        "executionTimeMillisEstimate": "300678.042"
    },
    "client": "172.31.5.63:53878",
    "appName": "MongoDB Shell",
    "user": "example"
}
```

Pour accélérer la requête dans cet exemple, vous souhaitez créer un index sur `fare.totalAmount`, comme indiqué ci-dessous.

```
db.rides.createIndex( {"fare.totalAmount": 1}, {background: true} )
```

> **Note**
>
> Les index créés au premier plan (c'est-à-dire si l'option `{background:true}` n'a pas été fournie lors de la création de l'index) prennent un verrou d'écriture exclusive, ce qui empêche les applications d'écrire des données dans la collection jusqu'à ce que la génération de l'index soit terminée. Soyez conscient de cet impact potentiel lors de la création d'index sur les clusters de production. Lors de la création d'index, nous vous recommandons de définir `{background:true}`.

En général, vous souhaitez créer des index sur des champs de cardinalité élevée (par exemple, un grand nombre de valeurs uniques). La création d'un index sur un champ à faible cardinalité peut générer un index de taille importante qui n'est pas utilisé. L'optimiseur de requêtes Amazon DocumentDB prend en compte la taille globale de la collection et la sélectivité des index lors de la création d'un plan de requête. Vous verrez parfois le processeur de requêtes sélectionner un `COLLSCAN`, même lorsqu'un index est présent. Cela se produit lorsque le processeur de requêtes estime que l'utilisation de l'index n'apportera pas un avantage au niveau des performances par rapport à l'analyse de l'ensemble de la collection. Si vous voulez forcer le processeur de requêtes à utiliser un index particulier, vous pouvez utiliser l'opérateur `hint()` comme indiqué ci-dessous.

```
db.collection.find().hint("indexName")
```

Résumé des requêtes utiles

Les requêtes suivantes peuvent être utiles pour surveiller les performances et l'utilisation des ressources dans Amazon DocumentDB.

- Utilisez la commande suivante pour afficher les statistiques relatives à une collection spécifique, notamment les compteurs d'opérations, les statistiques de cache, les statistiques d'accès et les statistiques de taille :

  ```
  db.collection.stats()
  ```

- Utilisez la commande suivante pour afficher les statistiques relatives à chaque index créé dans une collection, notamment la taille de l'index, les statistiques de cache spécifiques à l'index et les statistiques d'utilisation de l'index :

```
db.collection.aggregate([{$indexStats:{}}]).pretty()
```

- Utilisez la requête suivante pour répertorier toutes les activités.

```
db.adminCommand({currentOp: 1, $all: 1});
```

- Le code suivant répertorie toutes les requêtes de longue durée ou bloquées.

```
db.adminCommand({aggregate: 1,
                 pipeline: [{$currentOp: {}},
                            {$match: {$or: [{secs_running: {$gt: 10}},
                                            {WaitState: {$exists: true}}]}},
                            {$project: {_id:0,
                                        opid: 1,
                                        secs_running: 1,
                                        WaitState: 1,
                                        blockedOn: 1,
                                        command: 1}}],
                 cursor: {}
                });
```

- Le code suivant met fin à une requête.

```
db.adminCommand({killOp: 1, op: <opid of running or blocked query>});
```

- Utilisez le code suivant pour obtenir une vue agrégée de l'état du système.

```
db.adminCommand({aggregate: 1,
                 pipeline: [{$currentOp: {allUsers: true, idleConnections: true}},
                            {$group: {_id: {desc: "$desc", ns: "$ns", WaitState:
 "$WaitState"}, count: {$sum: 1}}}],
                 cursor: {}
                });
```

Référence API de gestion de clusters, instances et ressources Amazon DocumentDB

Cette section décrit les opérations de gestion des clusters, instances et ressources pour Amazon DocumentDB (avec compatibilité MongoDB) qui sont accessibles via HTTP, l'AWS Command Line Interface(AWS CLI), ou leAWSKIT SDK. Vous pouvez utiliser ces API pour créer, supprimer et modifier des clusters et des instances.

> **Important**
>
> Ces API sont utilisées uniquement pour gérer les clusters, les instances et les ressources connexes. Pour plus d'informations sur comment connecter un cluster Amazon DocumentDB en cours d'exécution, consultezGuide de démarrage.

Rubriques

Actions

Les actions suivantes sont prises en charge par Amazon DocumentDB (with MongoDB compatibility) :

- CreateDBClusterSnapshot
- CreateDBInstance
- CreateDBSubnetGroup
- CreateEventSubscription
- CreateGlobalCluster
- DeleteDBCluster
- DeleteDBClusterParameterGroup
- DeleteDBClusterSnapshot
- DeleteDBInstance
- DeleteDBSubnetGroup
- DeleteEventSubscription
- DeleteGlobalCluster
- DescribeCertificates
- DescribeDBClusterParameterGroups
- DescribeDBClusterParameters
- DescribeDBClusters
- DescribeDBClusterSnapshotAttributes
- DescribeDBClusterSnapshots
- DescribeDBEngineVersions
- DescribeDBInstances
- DescribeDBSubnetGroups
- DescribeEngineDefaultClusterParameters
- DescribeEventCategories
- DescribeEvents
- DescribeEventSubscriptions
- DescribeGlobalClusters
- DescribeOrderableDBInstanceOptions
- DescribePendingMaintenanceActions
- FailoverDBCluster
- ListTagsForResource

- ModifyDBCluster
- ModifyDBClusterParameterGroup
- ModifyDBClusterSnapshotAttribute
- ModifyDBInstance
- ModifyDBSubnetGroup
- ModifyEventSubscription
- ModifyGlobalCluster
- RebootDBInstance
- RemoveFromGlobalCluster
- RemoveSourceIdentifierFromSubscription
- RemoveTagsFromResource
- ResetDBClusterParameterGroup
- RestoreDBClusterFromSnapshot
- RestoreDBClusterToPointInTime
- StartDBCluster
- StopDBCluster

Les actions suivantes sont prises en charge par Amazon DocumentDB Elastic Clusters :

- CopyClusterSnapshot
- CreateCluster
- CreateClusterSnapshot
- DeleteCluster
- DeleteClusterSnapshot
- GetCluster
- GetClusterSnapshot
- ListClusters
- ListClusterSnapshots
- ListTagsForResource
- RestoreClusterFromSnapshot
- StartCluster

- StopCluster
- TagResource
- UntagResource
- UpdateCluster

Amazon DocumentDB (with MongoDB compatibility)

Les actions suivantes sont prises en charge parAmazon DocumentDB (with MongoDB compatibility) :

- AddSourceIdentifierToSubscription
- AddTagsToResource
- ApplyPendingMaintenanceAction
- CopyDBClusterParameterGroup
- CopyDBClusterSnapshot
- CreateDBCluster
- CreateDBClusterParameterGroup
- CreateDBClusterSnapshot
- CreateDBInstance
- CreateDBSubnetGroup
- CreateEventSubscription
- CreateGlobalCluster
- DeleteDBCluster
- DeleteDBClusterParameterGroup
- DeleteDBClusterSnapshot
- DeleteDBInstance
- DeleteDBSubnetGroup
- DeleteEventSubscription
- DeleteGlobalCluster
- DescribeCertificates
- DescribeDBClusterParameterGroups
- DescribeDBClusterParameters
- DescribeDBClusters

- DescribeDBClusterSnapshotAttributes
- DescribeDBClusterSnapshots
- DescribeDBEngineVersions
- DescribeDBInstances
- DescribeDBSubnetGroups
- DescribeEngineDefaultClusterParameters
- DescribeEventCategories
- DescribeEvents
- DescribeEventSubscriptions
- DescribeGlobalClusters
- DescribeOrderableDBInstanceOptions
- DescribePendingMaintenanceActions
- FailoverDBCluster
- ListTagsForResource
- ModifyDBCluster
- ModifyDBClusterParameterGroup
- ModifyDBClusterSnapshotAttribute
- ModifyDBInstance
- ModifyDBSubnetGroup
- ModifyEventSubscription
- ModifyGlobalCluster
- RebootDBInstance
- RemoveFromGlobalCluster
- RemoveSourceIdentifierFromSubscription
- RemoveTagsFromResource
- ResetDBClusterParameterGroup
- RestoreDBClusterFromSnapshot
- RestoreDBClusterToPointInTime
- StartDBCluster
- StopDBCluster

AddSourceIdentifierToSubscription

Service : Amazon DocumentDB (with MongoDB compatibility)

Ajoute un identifiant source à un abonnement à la notification d'événements existant.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

SourceIdentifier

Identifiant de la source d'événement à ajouter :

- Si le type de source est une instance, un `DBInstanceIdentifier` doit être fourni.
- Si le type de source est un groupe de sécurité, un `DBSecurityGroupName` doit être fourni.
- Si le type de source est un groupe de paramètres, un `DBParameterGroupName` doit être fourni.
- Si le type de source est un instantané, un `DBSnapshotIdentifier` doit être fourni.

Type : chaîne

Obligatoire : oui

SubscriptionName

Nom de l'abonnement aux notifications d'événements Amazon DocumentDB auquel vous souhaitez ajouter un identifiant de source.

Type : chaîne

Obligatoire : oui

Éléments de réponse

L'élément suivant est renvoyé par le service.

EventSubscription

Informations détaillées sur un événement auquel vous vous êtes inscrit.

Type : objet EventSubscription

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

SourceNotFound

La source demandée est introuvable.

Code d'état HTTP : 404

SubscriptionNotFound

Le nom de l'abonnement n'existe pas.

Code d'état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

AddTagsToResource

Service : Amazon DocumentDB (with MongoDB compatibility)

Ajoute des balises de métadonnées à une ressource Amazon DocumentDB. Vous pouvez utiliser ces balises avec les rapports de répartition des coûts pour suivre les coûts associés aux ressources Amazon DocumentDB ou dans une `Condition` déclaration d'une politique AWS Identity and Access Management (IAM) pour Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

ResourceName

La ressource Amazon DocumentDB à laquelle les balises sont ajoutées. Cette valeur est un nom de ressource Amazon.

Type : chaîne

Obligatoire : oui

Étiquettes.Tag.N

Les balises à attribuer à la ressource Amazon DocumentDB.

Type : tableau d’objets Tag

Obligatoire : oui

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterNotFoundFault

`DBClusterIdentifier`ne fait pas référence à un cluster existant.

Code d’état HTTP : 404

DBInstanceNotFound

`DBInstanceIdentifier`ne fait pas référence à une instance existante.

Code d'état HTTP : 404

DBSnapshotNotFound

`DBSnapshotIdentifier`ne fait pas référence à un instantané existant.

Code d'état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

ApplyPendingMaintenanceAction

Service : Amazon DocumentDB (with MongoDB compatibility)

Applique une action de maintenance en attente à une ressource (par exemple, à une instance Amazon DocumentDB).

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

ApplyAction

Action de maintenance en attente à appliquer à cette ressource.

Valeurs valides : `system-update`, `db-upgrade`

Type : chaîne

Obligatoire : oui

OptInType

Valeur qui spécifie le type de demande d'opt-in ou qui annule une demande d'opt-in. Un type demande de confirmation de l'acceptation de type `immediate` ne peut pas être annulée.

Valeurs valides :

- `immediate` - Appliquer immédiatement l'action de maintenance.
- `next-maintenance` - Appliquer l'action de maintenance pendant le créneau de maintenance suivant pour la ressource.
- `undo-opt-in` - Annuler toute demande de confirmation de l'acceptation `next-maintenance` existante.

Type : chaîne

Obligatoire : oui

ResourceIdentifier

ARN (Amazon Resource Name) de la ressource à laquelle s'applique l'action de maintenance en attente.

Type : chaîne

Obligatoire : oui

Éléments de réponse

L'élément suivant est renvoyé par le service.

ResourcePendingMaintenanceActions

Représente la sortie deApplyPendingMaintenanceAction.

Type : objet ResourcePendingMaintenanceActions

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

InvalidDBClusterStateFault

Le cluster n'est pas dans un état valide.

Code d’état HTTP : 400

InvalidDBInstanceState

L'instance spécifiée n'est pas dans l'état disponible.

Code d’état HTTP : 400

ResourceNotFoundFault

L'ID de ressource spécifiée est introuvable.

Code d’état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET

- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

CopyDBClusterParameterGroup

Service : Amazon DocumentDB (with MongoDB compatibility)

Copie le groupe de paramètres de cluster spécifié.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

SourceDBClusterParameterGroupIdentifier

Identifiant ou Amazon Resource Name (ARN) du groupe de paramètres du cluster source.

Contraintes :

- Vous devez spécifier un groupe de paramètres de cluster valide.
- Si le groupe de paramètres du cluster source est Région AWS identique à celui de la copie, spécifiez un identifiant de groupe de paramètres valide, par exemple`my-db-cluster-param-group`, ou un ARN valide.
- Si le groupe de paramètres source est différent de Région AWS celui de la copie, spécifiez un ARN de groupe de paramètres de cluster valide ; par exemple,`arn:aws:rds:us-east-1:123456789012:sample-cluster:sample-parameter-group`.

Type : chaîne

Obligatoire : oui

TargetDBClusterParameterGroupDescription

Description du groupe de paramètres de cluster copié.

Type : chaîne

Obligatoire : oui

TargetDBClusterParameterGroupIdentifier

Identifiant du groupe de paramètres de cluster copié.

Contraintes :

- Ne peut pas être null ou vide.
- Doit contenir entre 1 et 255 lettres, chiffres ou traits d'union.

- Le premier caractère doit être une lettre.
- Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

Exemple : `my-cluster-param-group1`

Type : chaîne

Obligatoire : oui

Étiquettes.Tag.N

Les balises qui doivent être attribuées au groupe de paramètres.

Type : tableau d'objets Tag

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBClusterParameterGroup

Informations détaillées sur un groupe de paramètres de cluster.

Type : objet DBClusterParameterGroup

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBParameterGroupAlreadyExists

Un groupe de paramètres portant le même nom existe déjà.

Code d'état HTTP : 400

DBParameterGroupNotFound

`DBParameterGroupName`ne fait pas référence à un groupe de paramètres existant.

Code d'état HTTP : 404

DBParameterGroupQuotaExceeded

Cette demande vous obligerait à dépasser le nombre autorisé de groupes de paramètres.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

CopyDBClusterSnapshot

Service : Amazon DocumentDB (with MongoDB compatibility)

Copie un instantané d'un cluster.

Pour copier un instantané de cluster à partir d'un instantané de cluster manuel partagé, `SourceDBClusterSnapshotIdentifier` il doit s'agir du nom de ressource Amazon (ARN) de l'instantané de cluster partagé. Vous ne pouvez copier qu'un instantané de cluster de bases de données partagé, chiffré ou non, dans la même Région AWS.

Pour annuler l'opération de copie une fois qu'elle est en cours, supprimez le cliché du cluster cible identifié `TargetDBClusterSnapshotIdentifier` alors que cet instantané de cluster est en état de copie.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

SourceDBClusterSnapshotIdentifier

L'identifiant de l'instantané du cluster à copier. Ce paramètre n'est pas sensible à la casse.

Contraintes :

- Vous devez spécifier un instantané système valide à l'état disponible.
- Si le cliché source est Région AWS identique à la copie, spécifiez un identifiant de cliché valide.
- Si l'instantané source se trouve dans un emplacement différent Région AWS de celui de la copie, spécifiez un ARN de capture de cluster valide.

Exemple : `my-cluster-snapshot1`

Type : chaîne

Obligatoire : oui

TargetDBClusterSnapshotIdentifier

L'identifiant du nouvel instantané du cluster à créer à partir de l'instantané du cluster source. Ce paramètre n'est pas sensible à la casse.

Contraintes :

- Doit contenir entre 1 et 63 lettres, chiffres ou traits d'union.
- Le premier caractère doit être une lettre.
- Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

Exemple : `my-cluster-snapshot2`

Type : chaîne

Obligatoire : oui

CopyTags

Définissez sur `true` pour copier toutes les balises de l'instantané du cluster source vers le cliché du cluster cible, et sinon`false`. L'argument par défaut est `false`.

Type : booléen

Obligatoire : non

KmsKeyId

ID de AWS KMS clé pour un instantané de cluster chiffré. L'ID de AWS KMS clé est le Amazon Resource Name (ARN), AWS KMS l'identifiant de AWS KMS clé ou l'alias de clé de AWS KMS chiffrement.

Si vous copiez un instantané de cluster chiffré depuis votre Compte AWS, vous pouvez spécifier une valeur `KmsKeyId` pour chiffrer la copie avec une nouvelle clé de AWS KMS chiffrement. Si vous ne spécifiez aucune valeur pour`KmsKeyId`, la copie du cliché de cluster est chiffrée avec la même AWS KMS clé que l'instantané de cluster source.

Si vous copiez un instantané de cluster chiffré partagé depuis un autre Compte AWS, vous devez spécifier une valeur pour`KmsKeyId`.

Pour copier un instantané de cluster chiffré vers un autre Région AWS, définissez `KmsKeyId` l'ID de AWS KMS clé que vous souhaitez utiliser pour chiffrer la copie de l'instantané de cluster dans la région de destination. AWS KMS les clés de chiffrement sont spécifiques à Région AWS celle dans laquelle elles ont été créées, et vous ne pouvez pas utiliser les clés de chiffrement les unes Région AWS des autres Région AWS.

Si vous copiez un instantané de cluster non chiffré et que vous spécifiez une valeur pour le `KmsKeyId` paramètre, une erreur est renvoyée.

Type : chaîne

Obligatoire : non

PreSignedUrl

L'URL qui contient une demande signée Signature Version 4 pour l'action d'`CopyDBClusterSnapshot`API dans Région AWS laquelle figure l'instantané du cluster source à copier. Vous devez utiliser le `PreSignedUrl` paramètre lorsque vous copiez un instantané de cluster à partir d'un autre Région AWS.

Si vous utilisez un outil AWS SDK ou le AWS CLI, vous pouvez le spécifier `SourceRegion` (ou `--source-region` pour le AWS CLI) au lieu de le spécifier `PreSignedUrl` manuellement. La spécification `SourceRegion` génère automatiquement une URL pré-signée qui est une demande valide pour l'opération pouvant être exécutée dans la source. Région AWS

L'URL présignée doit être une demande valide pour l'action d'`CopyDBClusterSnapshot`API qui peut être exécutée dans la source Région AWS contenant l'instantané du cluster à copier. La demande d'URL pré-signée doit contenir les valeurs de paramètres suivantes :

- `SourceRegion`- L'ID de la région qui contient le cliché à copier.
- `SourceDBClusterSnapshotIdentifier`- L'identifiant de l'instantané du cluster chiffré à copier. Cet identifiant doit être au format Amazon Resource Name (ARN) pour la Région AWS source. Par exemple, si vous copiez un instantané de cluster chiffré à partir de l' Région AWS us-east-1, cela ressemble à ce qui `SourceDBClusterSnapshotIdentifier` suit : `arn:aws:rds:us-east-1:12345678012:sample-cluster:sample-cluster-snapshot`
- `TargetDBClusterSnapshotIdentifier`- L'identifiant du nouvel instantané du cluster à créer. Ce paramètre n'est pas sensible à la casse.

Type : chaîne

Obligatoire : non

Étiquettes.Tag.N

Les balises à attribuer à l'instantané du cluster.

Type : tableau d’objets Tag

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBClusterSnapshot

Informations détaillées sur un instantané de cluster.

Type : objet DBClusterSnapshot

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterSnapshotAlreadyExistsFault

Vous disposez déjà d'un instantané du cluster avec l'identifiant indiqué.

Code d'état HTTP : 400

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`ne fait pas référence à un instantané de cluster existant.

Code d'état HTTP : 404

InvalidDBClusterSnapshotStateFault

La valeur fournie n'est pas un état de capture d'écran de cluster valide.

Code d'état HTTP : 400

InvalidDBClusterStateFault

Le cluster n'est pas dans un état valide.

Code d'état HTTP : 400

KMSKeyNotAccessibleFault

Une erreur s'est produite lors de l'accès à une AWS KMS clé.

Code d'état HTTP : 400

SnapshotQuotaExceeded

La demande vous obligerait à dépasser le nombre autorisé de clichés.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

CreateDBCluster

Service : Amazon DocumentDB (with MongoDB compatibility)

Crée un nouveau cluster Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterIdentifier

L'identifiant du cluster. Ce paramètre est stocké sous la forme d'une chaîne en lettres minuscules.

Contraintes :

- Doit contenir entre 1 et 63 lettres, chiffres ou traits d'union.
- Le premier caractère doit être une lettre.
- Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

Exemple : `my-cluster`

Type : chaîne

Obligatoire : oui

Engine

Nom du moteur de base de données à utiliser pour ce cluster.

Valeurs valides : `docdb`

Type : chaîne

Obligatoire : oui

AvailabilityZones. AvailabilityZoneN.

Liste des zones de disponibilité Amazon EC2 dans lesquelles les instances du cluster peuvent être créées.

Type : tableau de chaînes

Obligatoire : non

BackupRetentionPeriod

Nombre de jours de conservation des sauvegardes automatiques. Vous devez spécifier une valeur minimale de 1.

Par défaut : 1

Contraintes :

- Doit être une valeur comprise entre 1 et 35.

Type : entier

Obligatoire : non

DBClusterParameterGroupName

Nom du groupe de paramètres de cluster à associer à ce cluster.

Type : chaîne

Obligatoire : non

DBSubnetGroupName

Groupe de sous-réseaux à associer à ce cluster.

Contraintes : doit correspondre au nom d'un `DBSubnetGroup` existant. Impossible de conserver le nom par défaut.

Exemple : `mySubnetgroup`

Type : chaîne

Obligatoire : non

DeletionProtection

Spécifie si ce cluster peut être supprimé. Si cette option `DeletionProtection` est activée, le cluster ne peut pas être supprimé sauf s'il `DeletionProtection` est modifié et désactivé. `DeletionProtection`protège les clusters contre la suppression accidentelle.

Type : booléen

Obligatoire : non

EnableCloudwatchLogsExports.membre.n

Liste des types de journaux qui doivent être activés pour être exportés vers Amazon CloudWatch Logs. Vous pouvez activer les journaux d'audit ou les journaux de profileur. Pour plus d'informations, consultez Audit des événements Amazon DocumentDB et profilage des opérations Amazon DocumentDB.

Type : tableau de chaînes

Obligatoire : non

EngineVersion

Numéro de version du moteur de base de données à utiliser. La version `--engine-version` utilise par défaut la dernière version majeure du moteur. Pour les charges de travail de production, nous vous recommandons de déclarer explicitement ce paramètre avec la version majeur du moteur prévue.

Type : chaîne

Obligatoire : non

GlobalClusterIdentifier

Identifiant de cluster du nouveau cluster global.

Type : chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 255.

Modèle : `[A-Za-z][0-9A-Za-z-:._]*`

Obligatoire : non

KmsKeyId

Identifiant de AWS KMS clé pour un cluster chiffré.

L'identifiant de AWS KMS clé est le Amazon Resource Name (ARN) de la clé de AWS KMS chiffrement. Si vous créez un cluster à l'aide du même cluster Compte AWS qui possède la clé de AWS KMS chiffrement utilisée pour chiffrer le nouveau cluster, vous pouvez utiliser l'alias de AWS KMS clé au lieu de l'ARN pour la clé de AWS KMS chiffrement.

Si aucune clé de chiffrement n'est spécifiée dans `KmsKeyId` :

- Si le paramètre `StorageEncrypted` est `true`, Amazon DocumentDB utilise votre clé de chiffrement par défaut.

AWS KMS crée la clé de chiffrement par défaut pour votre Compte AWS. Vous disposez Compte AWS d'une clé de chiffrement par défaut différente pour chacune d'entre elles Régions AWS.

Type : chaîne

Obligatoire : non

MasterUsername

Nom de l'utilisateur principal du cluster.

Contraintes :

- Doit comporter entre 1 et 63 lettres ou chiffres.
- Le premier caractère doit être une lettre.
- Ne doit pas être un mot réservé pour le moteur de base de données choisi.

Type : chaîne

Obligatoire : non

MasterUserPassword

Mot de passe de l'utilisateur principal de la base de données. Ce mot de passe peut contenir tout caractère ASCII imprimable à l'exception de la barre oblique (/), des guillemets doubles (") ou du symbole arobase (@).

Contraintes : doit comporter entre 8 et 100 caractères.

Type : chaîne

Obligatoire : non

Port

Numéro de port sur lequel les instances du cluster acceptent les connexions.

Type : entier

Obligatoire : non

PreferredBackupWindow

Plage de temps quotidienne au cours de laquelle les sauvegardes automatiques sont créées si cette fonctionnalité est activée via le paramètre `BackupRetentionPeriod`.

La valeur par défaut est une fenêtre de 30 minutes sélectionnée au hasard dans un intervalle de 8 heures pour chacune d'entre elles. Région AWS

Contraintes :

- Doit être au format `hh24:mi-hh24:mi`.
- Doit être exprimée en heure UTC (Universal Coordinated Time).
- Ne doit pas être en conflit avec la fenêtre de maintenance préférée.
- Doit être de 30 minutes minimum.

Type : chaîne

Obligatoire : non

PreferredMaintenanceWindow

Intervalle de temps hebdomadaire, au format Universal Coordinated Time (UTC), pendant lequel a lieu la maintenance du système.

Format : `ddd:hh24:mi-ddd:hh24:mi`

La valeur par défaut est une fenêtre de 30 minutes sélectionnée au hasard dans un intervalle de 8 heures pour chacune d'elles Région AWS, survenant un jour aléatoire de la semaine.

Jours valides : Mon, Tue, Wed, Thu, Fri, Sat, Sun

Contraintes : fenêtre minimale de 30 minutes.

Type : chaîne

Obligatoire : non

PreSignedUrl

Cette option n'est pas prise en charge actuellement.

Type : chaîne

Obligatoire : non

StorageEncrypted

Indique si le cluster est chiffré.

Type : booléen

Obligatoire : non

StorageType

Type de stockage à associer au cluster de base de données.

Pour plus d'informations sur les types de stockage pour les clusters Amazon DocumentDB, consultez la section Configurations de stockage des clusters dans le manuel Amazon DocumentDB Developer Guide.

Valeurs valides pour le type de stockage - `standard | iopt1`

La valeur par défaut est `standard`

> **Note**
>
> Lorsque vous créez un cluster de base de données DocumentDB avec le type de stockage défini sur`iopt1`, le type de stockage est renvoyé dans la réponse. Le type de stockage n'est pas renvoyé lorsque vous le définissez sur`standard`.

Type : chaîne

Obligatoire : non

Étiquettes.Tag.N

Balises à attribuer au cluster.

Type : tableau d’objets Tag

Obligatoire : non

VpcSecurityGroupIds. VpcSecurityGroupIdN.

Liste des groupes de sécurité VPC EC2 à associer à ce cluster.

Type : tableau de chaînes

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBCluster

Informations détaillées sur un cluster.

Type : objet DBCluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterAlreadyExistsFault

Vous avez déjà un cluster avec l'identifiant indiqué.

Code d'état HTTP : 400

DBClusterNotFoundFault

`DBClusterIdentifier`ne fait pas référence à un cluster existant.

Code d'état HTTP : 404

DBClusterParameterGroupNotFound

`DBClusterParameterGroupName`ne fait pas référence à un groupe de paramètres de cluster existant.

Code d'état HTTP : 404

DBClusterQuotaExceededFault

Le cluster ne peut pas être créé car vous avez atteint le quota maximum autorisé de clusters.

Code d'état HTTP : 403

DBInstanceNotFound

`DBInstanceIdentifier`ne fait pas référence à une instance existante.

Code d’état HTTP : 404

DBSubnetGroupDoesNotCoverEnoughAZs

Les sous-réseaux du groupe de sous-réseaux doivent couvrir au moins deux zones de disponibilité, sauf s'il n'existe qu'une seule zone de disponibilité.

Code d’état HTTP : 400

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`ne fait pas référence à un groupe de sous-réseaux existant.

Code d’état HTTP : 404

GlobalClusterNotFoundFault

`GlobalClusterIdentifier`Cela ne fait pas référence à un cluster mondial existant.

Code d’état HTTP : 404

InsufficientStorageClusterCapacity

L'espace de stockage disponible est insuffisant pour l'action en cours. Vous pouvez peut-être résoudre cette erreur en mettant à jour votre groupe de sous-réseaux afin qu'il utilise différentes zones de disponibilité disposant d'un espace de stockage plus important.

Code d’état HTTP : 400

InvalidDBClusterStateFault

Le cluster n'est pas dans un état valide.

Code d’état HTTP : 400

InvalidDBInstanceState

L'instance spécifiée n'est pas dans l'état disponible.

Code d’état HTTP : 400

InvalidDBSubnetGroupStateFault

Le groupe de sous-réseaux ne peut pas être supprimé car il est en cours d'utilisation.

Code d’état HTTP : 400

InvalidGlobalClusterStateFault

L'opération demandée ne peut pas être effectuée tant que le cluster est dans cet état.

Code d'état HTTP : 400

InvalidSubnet

Le sous-réseau demandé n'est pas valide ou plusieurs sous-réseaux ont été demandés mais ils ne se trouvent pas tous dans un cloud privé virtuel (VPC) commun.

Code d'état HTTP : 400

InvalidVPCNetworkStateFault

Le groupe de sous-réseaux ne couvre pas toutes les zones de disponibilité après sa création en raison des modifications apportées.

Code d'état HTTP : 400

KMSKeyNotAccessibleFault

Une erreur s'est produite lors de l'accès à une AWS KMS clé.

Code d'état HTTP : 400

StorageQuotaExceeded

La demande vous obligerait à dépasser la quantité de stockage autorisée disponible sur toutes les instances.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2

- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

CreateDBClusterParameterGroup

Service : Amazon DocumentDB (with MongoDB compatibility)

Crée un nouveau groupe de paramètres de cluster.

Les paramètres d'un groupe de paramètres de cluster s'appliquent à toutes les instances d'un cluster.

Un groupe de paramètres de cluster est initialement créé avec les paramètres par défaut pour le moteur de base de données utilisé par les instances du cluster. Dans Amazon DocumentDB, vous ne pouvez pas modifier directement le groupe de paramètres du `default.docdb3.6` cluster. Si votre cluster Amazon DocumentDB utilise le groupe de paramètres de cluster par défaut et que vous souhaitez y modifier une valeur, vous devez d'abord créer un nouveau groupe de paramètres ou copier un groupe de paramètres existant, le modifier, puis appliquer le groupe de paramètres modifié à votre cluster. Pour que le nouveau groupe de paramètres de cluster et les paramètres associés prennent effet, vous devez ensuite redémarrer les instances du cluster sans basculement. Pour plus d'informations, consultez Modifier les groupes de paramètres du cluster Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterParameterGroupName

Nom du groupe de paramètres de cluster.

Contraintes :

- Ne doit pas correspondre au nom d'un élément `DBClusterParameterGroup` existant.

> **Note**
>
> Cette valeur est stockée sous la forme d'une chaîne en minuscules.

Type : chaîne

Obligatoire : oui

DBParameterGroupFamily

Nom de la famille du groupe de paramètres de cluster.

Type : chaîne

Obligatoire : oui

Description

Description du groupe de paramètres de cluster.

Type : chaîne

Obligatoire : oui

Étiquettes.Tag.N

Balises à attribuer au groupe de paramètres de cluster.

Type : tableau d'objets Tag

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBClusterParameterGroup

Informations détaillées sur un groupe de paramètres de cluster.

Type : objet DBClusterParameterGroup

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBParameterGroupAlreadyExists

Un groupe de paramètres portant le même nom existe déjà.

Code d'état HTTP : 400

DBParameterGroupQuotaExceeded

Cette demande vous obligerait à dépasser le nombre autorisé de groupes de paramètres.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

CreateDBClusterSnapshot

Service : Amazon DocumentDB (with MongoDB compatibility)

Crée un instantané d'un cluster.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterIdentifier

Identifiant du cluster pour lequel créer un instantané. Ce paramètre n'est pas sensible à la casse.

Contraintes :

- Doit correspondre à l'identifiant d'un `DBCluster` existant.

Exemple : `my-cluster`

Type : chaîne

Obligatoire : oui

DBClusterSnapshotIdentifier

Identifiant du snapshot du cluster. Ce paramètre est stocké sous la forme d'une chaîne en lettres minuscules.

Contraintes :

- Doit contenir entre 1 et 63 lettres, chiffres ou traits d'union.
- Le premier caractère doit être une lettre.
- Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

Exemple : `my-cluster-snapshot1`

Type : chaîne

Obligatoire : oui

Étiquettes.Tag.N

Les balises à attribuer à l'instantané du cluster.

Type : tableau d’objets Tag

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBClusterSnapshot

Informations détaillées sur un instantané de cluster.

Type : objet DBClusterSnapshot

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterNotFoundFault

`DBClusterIdentifier`ne fait pas référence à un cluster existant.

Code d'état HTTP : 404

DBClusterSnapshotAlreadyExistsFault

Vous disposez déjà d'un instantané du cluster avec l'identifiant indiqué.

Code d'état HTTP : 400

InvalidDBClusterSnapshotStateFault

La valeur fournie n'est pas un état de capture d'écran de cluster valide.

Code d'état HTTP : 400

InvalidDBClusterStateFault

Le cluster n'est pas dans un état valide.

Code d'état HTTP : 400

SnapshotQuotaExceeded

La demande vous obligerait à dépasser le nombre autorisé de clichés.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

CreateDBInstance

Service : Amazon DocumentDB (with MongoDB compatibility)

Crée une nouvelle instance.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterIdentifier

Identifiant du cluster auquel l'instance appartiendra.

Type : chaîne

Obligatoire : oui

DBInstanceClass

Capacité de calcul et de mémoire de l'instance ; par exemple, `db.r5.large`.

Type : chaîne

Obligatoire : oui

DBInstanceIdentifier

Identifiant de l’instance. Ce paramètre est stocké sous la forme d'une chaîne en lettres minuscules.

Contraintes :

- Doit contenir entre 1 et 63 lettres, chiffres ou traits d'union.
- Le premier caractère doit être une lettre.
- Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

Exemple : `mydbinstance`

Type : chaîne

Obligatoire : oui

Engine

Nom du moteur de base de données à utiliser pour cette instance.

Valeur valide : `docdb`

Type : chaîne

Obligatoire : oui

AutoMinorVersionUpgrade

Ce paramètre ne s'applique pas à Amazon DocumentDB. Amazon DocumentDB n'effectue pas de mises à niveau mineures de version, quelle que soit la valeur définie.

Par défaut : `false`

Type : booléen

Obligatoire : non

AvailabilityZone

Zone de disponibilité Amazon EC2 dans laquelle l'instance est créée.

Par défaut : une zone de disponibilité choisie au hasard par le système dans le terminal. Région AWS

Exemple : `us-east-1d`

Type : chaîne

Obligatoire : non

CACertificateIdentifier

L'identifiant de certificat CA à utiliser pour le certificat de serveur de l'instance de base de données.

Pour plus d'informations, consultez la section Mise à jour de vos certificats TLS Amazon DocumentDB et chiffrement des données en transit dans le guide du développeur Amazon DocumentDB.

Type : chaîne

Obligatoire : non

CopyTagsToSnapshot

Une valeur qui indique si vous voulez copier toutes les balises à partir de l'instance de base de données pour les instantanés de l'instance de base de données. Par défaut, les balises ne sont pas copiées.

Type : booléen

Obligatoire : non

EnablePerformanceInsights

Une valeur qui indique s'il convient d'activer Performance Insights pour l'instance de base de données. Pour plus d'informations, voir Utilisation d'Amazon Performance Insights.

Type : booléen

Obligatoire : non

PerformanceInsightsKMSKeyId

Identifiant AWS KMS clé pour le chiffrement des données Performance Insights.

L'identifiant de AWS KMS clé est l'ARN de la clé, l'ID de clé, l'alias ARN ou le nom d'alias de la clé KMS.

Si vous ne spécifiez aucune valeur pour PerformanceInsights KMSKeyId, Amazon DocumentDB utilise votre clé KMS par défaut. Il existe une clé KMS par défaut pour votre compte Amazon Web Services. Votre compte Amazon Web Services possède une clé KMS par défaut différente pour chaque région Amazon Web Services.

Type : chaîne

Obligatoire : non

PreferredMaintenanceWindow

Intervalle de temps hebdomadaire, au format UTC (temps universel), pendant lequel a lieu la maintenance du système.

Format : `ddd:hh24:mi-ddd:hh24:mi`

La valeur par défaut est une fenêtre de 30 minutes sélectionnée au hasard dans un intervalle de 8 heures pour chacune d'entre elles Région AWS, survenant un jour aléatoire de la semaine.

Jours valides : Mon, Tue, Wed, Thu, Fri, Sat, Sun

Contraintes : fenêtre minimale de 30 minutes.

Type : chaîne

Obligatoire : non

PromotionTier

Valeur qui indique l'ordre dans lequel une réplique Amazon DocumentDB est promue vers l'instance principale après une défaillance de l'instance principale existante.

Valeur par défaut : 1

Valeurs valides : 0 à 15

Type : entier

Obligatoire : non

Étiquettes.Tag.N

Balises à attribuer à l'instance. Vous pouvez attribuer jusqu'à 10 balises à une instance.

Type : tableau d'objets Tag

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBInstance

Informations détaillées sur une instance.

Type : objet DBInstance

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

AuthorizationNotFound

L'adresse IP CIDR ou le groupe de sécurité Amazon EC2 spécifié n'est pas autorisé pour le groupe de sécurité spécifié.

Amazon DocumentDB peut également ne pas être autorisé à effectuer les actions nécessaires en votre nom à l'aide d'IAM.

Code d'état HTTP : 404

DBClusterNotFoundFault

`DBClusterIdentifier`ne fait pas référence à un cluster existant.

Code d'état HTTP : 404

DBInstanceAlreadyExists

Vous avez déjà une instance avec l'identifiant indiqué.

Code d'état HTTP : 400

DBParameterGroupNotFound

`DBParameterGroupName`ne fait pas référence à un groupe de paramètres existant.

Code d'état HTTP : 404

DBSecurityGroupNotFound

`DBSecurityGroupName`ne fait pas référence à un groupe de sécurité existant.

Code d'état HTTP : 404

DBSubnetGroupDoesNotCoverEnoughAZs

Les sous-réseaux du groupe de sous-réseaux doivent couvrir au moins deux zones de disponibilité, sauf s'il n'existe qu'une seule zone de disponibilité.

Code d'état HTTP : 400

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`ne fait pas référence à un groupe de sous-réseaux existant.

Code d'état HTTP : 404

InstanceQuotaExceeded

La demande vous obligerait à dépasser le nombre d'instances autorisé.

Code d'état HTTP : 400

InsufficientDBInstanceCapacity

La classe d'instance spécifiée n'est pas disponible dans la zone de disponibilité spécifiée.

Code d'état HTTP : 400

InvalidDBClusterStateFault

Le cluster n'est pas dans un état valide.

Code d'état HTTP : 400

InvalidSubnet

Le sous-réseau demandé n'est pas valide ou plusieurs sous-réseaux ont été demandés mais ils ne se trouvent pas tous dans un cloud privé virtuel (VPC) commun.

Code d'état HTTP : 400

InvalidVPCNetworkStateFault

Le groupe de sous-réseaux ne couvre pas toutes les zones de disponibilité après sa création en raison des modifications apportées.

Code d'état HTTP : 400

KMSKeyNotAccessibleFault

Une erreur s'est produite lors de l'accès à une AWS KMS clé.

Code d'état HTTP : 400

StorageQuotaExceeded

La demande vous obligerait à dépasser la quantité de stockage autorisée disponible sur toutes les instances.

Code d'état HTTP : 400

StorageTypeNotSupported

Le stockage du paramètre spécifié ne `StorageType` peut pas être associé à l'instance de base de données.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

CreateDBSubnetGroup

Service : Amazon DocumentDB (with MongoDB compatibility)

Crée un nouveau groupe de sous-réseaux. Les groupes de sous-réseaux doivent contenir au moins un sous-réseau dans au moins deux zones de disponibilité du. Région AWS

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBSubnetGroupDescription

Description du groupe de sous-réseaux.

Type : chaîne

Obligatoire : oui

DBSubnetGroupName

Nom du groupe de sous-réseaux. Cette valeur est stockée sous la forme d'une chaîne en minuscules.

Contraintes : doit comporter au maximum 255 lettres, chiffres, points, traits de soulignement, espaces ou tirets. Impossible de conserver le nom par défaut.

Exemple : `mySubnetgroup`

Type : chaîne

Obligatoire : oui

SubnetIds. SubnetIdentifierN.

ID de sous-réseau Amazon EC2 du groupe de sous-réseaux.

Type : tableau de chaînes

Obligatoire : oui

Étiquettes.Tag.N

Balises à attribuer au groupe de sous-réseaux.

Type : tableau d’objets Tag

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBSubnetGroup

Informations détaillées sur un groupe de sous-réseaux.

Type : objet DBSubnetGroup

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBSubnetGroupAlreadyExists

`DBSubnetGroupName`est déjà utilisé par un groupe de sous-réseaux existant.

Code d’état HTTP : 400

DBSubnetGroupDoesNotCoverEnoughAZs

Les sous-réseaux du groupe de sous-réseaux doivent couvrir au moins deux zones de disponibilité, sauf s'il n'existe qu'une seule zone de disponibilité.

Code d’état HTTP : 400

DBSubnetGroupQuotaExceeded

La demande vous obligerait à dépasser le nombre autorisé de groupes de sous-réseaux.

Code d’état HTTP : 400

DBSubnetQuotaExceededFault

La demande vous obligerait à dépasser le nombre autorisé de sous-réseaux dans un groupe de sous-réseaux.

Code d’état HTTP : 400

InvalidSubnet

Le sous-réseau demandé n'est pas valide ou plusieurs sous-réseaux ont été demandés mais ils ne se trouvent pas tous dans un cloud privé virtuel (VPC) commun.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

CreateEventSubscription

Service : Amazon DocumentDB (with MongoDB compatibility)

Crée un abonnement aux notifications d'événements Amazon DocumentDB. Cette action nécessite un nom de ressource Amazon (ARN) de rubrique créé à l'aide de la console Amazon DocumentDB, de la console Amazon SNS ou de l'API Amazon SNS. Pour obtenir un ARN avec Amazon SNS, vous devez créer une rubrique dans Amazon SNS et vous y abonner. L'ARN est affiché dans la console Amazon SNS.

Vous pouvez spécifier le type de source (`SourceType`) dont vous souhaitez être informé. Vous pouvez également fournir une liste des sources Amazon DocumentDB (`SourceIds`) qui déclenchent les événements, ainsi qu'une liste de catégories d'événements (`EventCategories`) pour les événements dont vous souhaitez être informé. Par exemple, vous pouvez spécifier`SourceType = db-instance`, `SourceIds = mydbinstance1, mydbinstance2` et`EventCategories = Availability, Backup`.

Si vous spécifiez à la fois le `SourceType` et `SourceIds` (tel que `SourceType = db-instance` et`SourceIdentifier = myDBInstance1`), vous êtes informé de tous les `db-instance` événements relatifs à la source spécifiée. Si vous spécifiez a `SourceType` mais pas a`SourceIdentifier`, vous êtes informé des événements associés à ce type de source pour toutes vos sources Amazon DocumentDB. Si vous ne spécifiez `SourceType` ni le`SourceIdentifier`, vous êtes informé des événements générés par toutes les sources Amazon DocumentDB appartenant à votre compte client.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

SnsTopicArn

Amazon Resource Name (ARN) de la rubrique SNS créé pour la notification d'événements. Amazon SNS crée l'ARN lorsque vous créez une rubrique et que vous vous y abonnez.

Type : chaîne

Obligatoire : oui

SubscriptionName

Nom de l'abonnement.

Contraintes : Le nom doit comporter moins de 255 caractères.

Type : chaîne

Obligatoire : oui

Enabled

Valeur booléenne ; définie sur `true` pour activer l'abonnement, définie pour `false` créer l'abonnement mais pas pour l'activer.

Type : booléen

Obligatoire : non

EventCategories. EventCategoryN.

Liste des catégories d'événements `SourceType` auxquels vous souhaitez vous abonner.

Type : tableau de chaînes

Obligatoire : non

SourceIds. SourceIdN.

Liste des identifiants des sources d'événements pour lesquels des événements sont renvoyés. Si la valeur n'est pas spécifiée, toutes les sources sont incluses dans la réponse. Un identifiant doit commencer par une lettre et contenir uniquement des lettres ASCII, des chiffres et des tirets. Il ne doit pas se terminer par un tiret ou contenir deux tirets consécutifs.

Contraintes :

- S'`SourceIds`ils sont fournis, ils `SourceType` doivent également être fournis.
- Si le type de source est une instance, un `DBInstanceIdentifier` doit être fourni.
- Si le type de source est un groupe de sécurité, un `DBSecurityGroupName` doit être fourni.
- Si le type de source est un groupe de paramètres, un `DBParameterGroupName` doit être fourni.
- Si le type de source est un instantané, un `DBSnapshotIdentifier` doit être fourni.

Type : tableau de chaînes

Obligatoire : non

SourceType

Type de source qui génère les événements. Par exemple, si vous souhaitez être informé des événements générés par une instance, vous devez définir ce paramètre sur`db-instance`. Si cette valeur n'est pas spécifiée, tous les événements sont renvoyés.

Valeurs valides: `db-instance`, `db-cluster`, `db-parameter-group`, `db-security-group`, `db-cluster-snapshot`

Type : chaîne

Obligatoire : non

Étiquettes.Tag.N

Les tags à attribuer à l'abonnement à l'événement.

Type : tableau d’objets Tag

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

EventSubscription

Informations détaillées sur un événement auquel vous vous êtes inscrit.

Type : objet EventSubscription

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

EventSubscriptionQuotaExceeded

Vous avez atteint le nombre maximum d'abonnements aux événements.

Code d’état HTTP : 400

SNSInvalidTopic

Amazon SNS a répondu qu'il y avait un problème avec le sujet spécifié.

Code d'état HTTP : 400

SNSNoAuthorization

Vous n'êtes pas autorisé à publier sur la rubrique SNS Amazon Resource Name (ARN).

Code d'état HTTP : 400

SNSTopicArnNotFound

La rubrique SNS Amazon Resource Name (ARN) n'existe pas.

Code d'état HTTP : 404

SourceNotFound

La source demandée est introuvable.

Code d'état HTTP : 404

SubscriptionAlreadyExist

Le nom d'abonnement fourni existe déjà.

Code d'état HTTP : 400

SubscriptionCategoryNotFound

La catégorie fournie n'existe pas.

Code d'état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2

- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

CreateGlobalCluster

Service : Amazon DocumentDB (with MongoDB compatibility)

Crée un cluster global Amazon DocumentDB qui peut s'étendre sur plusieurs. Régions AWS Le cluster global contient un cluster principal doté d'une capacité de lecture-écriture, et jusqu'à des clusters secondaires en lecture seule. Les clusters mondiaux utilisent une réplication rapide basée sur le stockage entre les régions avec des latences inférieures à une seconde, en utilisant une infrastructure dédiée sans impact sur les performances de votre charge de travail.

Vous pouvez créer un cluster global initialement vide, puis y ajouter un cluster principal et un cluster secondaire. Vous pouvez également spécifier un cluster existant lors de l'opération de création, et ce cluster devient le principal du cluster global.

> **Note**
>
> Cette action s'applique uniquement aux clusters Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

GlobalClusterIdentifier

Identifiant de cluster du nouveau cluster global.

Type : chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 255.

Modèle : `[A-Za-z][0-9A-Za-z-:._]*`

Obligatoire : oui

DatabaseName

Nom de votre base de données comprenant au maximum 64 caractères alphanumériques. Si vous ne fournissez pas de nom, Amazon DocumentDB ne créera pas de base de données dans le cluster global que vous créez.

Type : chaîne

Obligatoire : non

DeletionProtection

Le paramètre de protection contre la suppression pour le nouveau cluster global. Le cluster global ne peut pas être supprimé lorsque la protection contre la suppression est activée.

Type : booléen

Obligatoire : non

Engine

Nom du moteur de base de données à utiliser pour ce cluster.

Type : chaîne

Obligatoire : non

EngineVersion

Version du moteur du cluster global.

Type : chaîne

Obligatoire : non

SourceDBClusterIdentifier

Le nom de ressource Amazon (ARN) à utiliser comme cluster principal du cluster mondial. Ce paramètre est facultatif.

Type : chaîne

Obligatoire : non

StorageEncrypted

Le paramètre de chiffrement du stockage pour le nouveau cluster global.

Type : booléen

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

GlobalCluster

Type de données représentant un cluster global Amazon DocumentDB.

Type : objet GlobalCluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterNotFoundFault

`DBClusterIdentifier`ne fait pas référence à un cluster existant.

Code d'état HTTP : 404

GlobalClusterAlreadyExistsFault

`GlobalClusterIdentifier` existe déjà. Choisissez un nouvel identifiant de cluster global (nom unique) pour créer un nouveau cluster global.

Code d'état HTTP : 400

GlobalClusterQuotaExceededFault

Le nombre de clusters globaux pour ce compte est déjà au maximum autorisé.

Code d'état HTTP : 400

InvalidDBClusterStateFault

Le cluster n'est pas dans un état valide.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DeleteDBCluster

Service : Amazon DocumentDB (with MongoDB compatibility)

Supprime un cluster précédemment provisionné. Lorsque vous supprimez un cluster, toutes les sauvegardes automatiques de ce cluster sont supprimées et ne peuvent pas être restaurées. Les instantanés manuels du cluster de base de données du cluster spécifié ne sont pas supprimés.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterIdentifier

Identifiant du cluster à supprimer. Ce paramètre n'est pas sensible à la casse.

Contraintes :

- Doit correspondre à un existant`DBClusterIdentifier`.

Type : chaîne

Obligatoire : oui

FinalDBSnapshotIdentifier

L'identifiant de capture d'écran de cluster du nouveau cliché de cluster créé lorsque `SkipFinalSnapshot` ce paramètre est défini sur`false`.

> **Note**
>
> Le fait de spécifier ce paramètre et de le `SkipFinalShapshot` définir également sur `true` entraîne une erreur.

Contraintes :

- Il doit comporter de 1 à 255 lettres, chiffres ou traits d'union.
- Le premier caractère doit être une lettre.
- Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

Type : chaîne

Obligatoire : non

SkipFinalSnapshot

Détermine si un instantané final du cluster est créé avant la suppression du cluster. Si cette `true` option est spécifiée, aucun instantané de cluster n'est créé. Si cela `false` est spécifié, un instantané de cluster est créé avant que le cluster de base de données ne soit supprimé.

> **Note**
>
> Dans `SkipFinalSnapshot` l'`false`affirmative, vous devez spécifier un `FinalDBSnapshotIdentifier` paramètre.

Par défaut : `false`

Type : booléen

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBCluster

Informations détaillées sur un cluster.

Type : objet DBCluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterNotFoundFault

`DBClusterIdentifier`ne fait pas référence à un cluster existant.

Code d'état HTTP : 404

DBClusterSnapshotAlreadyExistsFault

Vous disposez déjà d'un instantané du cluster avec l'identifiant indiqué.

Code d’état HTTP : 400

InvalidDBClusterSnapshotStateFault

La valeur fournie n'est pas un état de capture d'écran de cluster valide.

Code d’état HTTP : 400

InvalidDBClusterStateFault

Le cluster n'est pas dans un état valide.

Code d’état HTTP : 400

SnapshotQuotaExceeded

La demande vous obligerait à dépasser le nombre autorisé de clichés.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DeleteDBClusterParameterGroup

Service : Amazon DocumentDB (with MongoDB compatibility)

Supprime un groupe de paramètres de cluster spécifié. Le groupe de paramètres de cluster à supprimer ne peut être associé à aucun cluster.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterParameterGroupName

Nom du groupe de paramètres de cluster.

Contraintes :

- Doit être le nom d'un groupe de paramètres de cluster existant.
- Vous ne pouvez pas supprimer un groupe de paramètres de cluster par défaut.
- Ne peut être associé à aucun cluster.

Type : chaîne

Obligatoire : oui

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBParameterGroupNotFound

`DBParameterGroupName`ne fait pas référence à un groupe de paramètres existant.

Code d'état HTTP : 404

InvalidDBParameterGroupState

Le groupe de paramètres est en cours d'utilisation ou son état n'est pas valide. Si vous essayez de supprimer le groupe de paramètres, vous ne pouvez pas le supprimer lorsque le groupe de paramètres est dans cet état.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DeleteDBClusterSnapshot

Service : Amazon DocumentDB (with MongoDB compatibility)

Supprime un instantané de cluster. Si l'instantané est en cours de copie, l'opération est arrêtée.

> **Note**
>
> L'instantané du cluster doit être dans l'`available`état pour être supprimé.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterSnapshotIdentifier

Identifiant du snapshot du cluster à supprimer.

Contraintes : Doit être le nom d'un instantané de cluster existant dans l'`available`état.

Type : chaîne

Obligatoire : oui

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBClusterSnapshot

Informations détaillées sur un instantané de cluster.

Type : objet DBClusterSnapshot

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`ne fait pas référence à un instantané de cluster existant.

Code d'état HTTP : 404

InvalidDBClusterSnapshotStateFault

La valeur fournie n'est pas un état de capture d'écran de cluster valide.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DeleteDBInstance

Service : Amazon DocumentDB (with MongoDB compatibility)

Supprime une instance précédemment provisionnée.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBInstanceIdentifier

L'identifiant de l'instance à supprimer. Ce paramètre n'est pas sensible à la casse.

Contraintes :

- Doit correspondre au nom d'une instance existante.

Type : chaîne

Obligatoire : oui

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBInstance

Informations détaillées sur une instance.

Type : objet DBInstance

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBInstanceNotFound

`DBInstanceIdentifier`ne fait pas référence à une instance existante.

Code d'état HTTP : 404

DBSnapshotAlreadyExists

`DBSnapshotIdentifier`est déjà utilisé par un instantané existant.

Code d'état HTTP : 400

InvalidDBClusterStateFault

L'état du cluster n'est pas valide.

Code d'état HTTP : 400

InvalidDBInstanceState

L'instance spécifiée n'est pas disponible.

Code d'état HTTP : 400

SnapshotQuotaExceeded

La demande vous obligerait à dépasser le nombre autorisé de clichés.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DeleteDBSubnetGroup

Service : Amazon DocumentDB (with MongoDB compatibility)

Supprime un groupe de sous-réseaux.

> **Note**
>
> Le groupe de sous-réseaux de base de données spécifié ne doit pas être associé à des instances de base de données.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBSubnetGroupName

Nom du groupe de sous-réseaux de base de données à supprimer.

> **Note**
>
> Vous ne pouvez pas supprimer le groupe de sous-réseaux par défaut.

Contraintes :

Doit correspondre au nom d'un `DBSubnetGroup` existant. Impossible de conserver le nom par défaut.

Exemple : `mySubnetgroup`

Type : chaîne

Obligatoire : oui

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`ne fait pas référence à un groupe de sous-réseaux existant.

Code d'état HTTP : 404

InvalidDBSubnetGroupStateFault

Le groupe de sous-réseaux ne peut pas être supprimé car il est en cours d'utilisation.

Code d'état HTTP : 400

InvalidDBSubnetStateFault

Le sous-réseau n'est pas dans l'état disponible.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DeleteEventSubscription

Service : Amazon DocumentDB (with MongoDB compatibility)

Supprime un abonnement aux notifications d'événements Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

SubscriptionName

Nom de l'abonnement aux notifications d'événements Amazon DocumentDB que vous souhaitez supprimer.

Type : chaîne

Obligatoire : oui

Éléments de réponse

L'élément suivant est renvoyé par le service.

EventSubscription

Informations détaillées sur un événement auquel vous vous êtes inscrit.

Type : objet EventSubscription

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

InvalidEventSubscriptionState

Quelqu'un d'autre est peut-être en train de modifier un abonnement. Patientez quelques secondes, puis réessayez.

Code d’état HTTP : 400

SubscriptionNotFound

Le nom de l'abonnement n'existe pas.

Code d’état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DeleteGlobalCluster

Service : Amazon DocumentDB (with MongoDB compatibility)

Supprime un cluster global. Les clusters principal et secondaire doivent déjà être détachés ou supprimés avant de tenter de supprimer un cluster global.

> **Note**
>
> Cette action s'applique uniquement aux clusters Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

GlobalClusterIdentifier

Identifiant du cluster global en cours de suppression.

Type : chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 255.

Modèle : `[A-Za-z][0-9A-Za-z-:._]*`

Obligatoire : oui

Éléments de réponse

L'élément suivant est renvoyé par le service.

GlobalCluster

Type de données représentant un cluster global Amazon DocumentDB.

Type : objet GlobalCluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

GlobalClusterNotFoundFault

`GlobalClusterIdentifier`Cela ne fait pas référence à un cluster mondial existant.

Code d'état HTTP : 404

InvalidGlobalClusterStateFault

L'opération demandée ne peut pas être effectuée tant que le cluster est dans cet état.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribeCertificates

Service : Amazon DocumentDB (with MongoDB compatibility)

Renvoie une liste des certificats d'autorité de certification (CA) fournis par Amazon DocumentDB à cet effet. Compte AWS

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

CertificateIdentifier

L'identifiant de certificat fourni par l'utilisateur. Si ce paramètre est spécifié, seules les informations relatives au certificat spécifié sont renvoyées. Si ce paramètre est omis, une liste de `MaxRecords` certificats maximum est renvoyée. Ce paramètre n'est pas sensible à la casse.

Constraints

- Doit correspondre à un existant`CertificateIdentifier`.

Type : chaîne

Obligatoire : non

Filtres.Filter.N

Ce paramètre n'est actuellement pas pris en charge.

Type : tableau d’objets Filter

Obligatoire : non

Marker

Jeton de pagination facultatif fourni par une demande `DescribeCertificates` précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Obligatoire : non

MaxRecords

Nombre maximal d'enregistrements à inclure dans la réponse. Si le nombre d'enregistrements existants est supérieur à la valeur `MaxRecords` spécifiée, un jeton de pagination appelé marqueur est inclus dans la réponse pour permettre la récupération des résultats restants.

Par défaut : 100

Contraintes :

- Minimum : 20
- Maximum : 100

Type : entier

Obligatoire : non

Éléments de réponse

Les éléments suivants sont renvoyés par le service.

Certificats.Certificat.N

Une liste de certificats pour cela Compte AWS.

Type : tableau d’objets Certificate

Marker

Un jeton de pagination facultatif fourni si le nombre d'enregistrements récupérés est supérieur à`MaxRecords`. Si ce paramètre est spécifié, le marqueur indique l'enregistrement suivant de la liste. Inclure la valeur de `Marker` dans le prochain appel aux `DescribeCertificates` résultats dans la page de certificats suivante.

Type : chaîne

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

CertificateNotFound

`CertificateIdentifier`ne fait pas référence à un certificat existant.

Code d'état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribeDBClusterParameterGroups

Service : Amazon DocumentDB (with MongoDB compatibility)

Renvoie une liste des descriptions de `DBClusterParameterGroup`. Si un `DBClusterParameterGroupName` paramètre est spécifié, la liste contient uniquement la description du groupe de paramètres de cluster spécifié.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterParameterGroupName

Nom d'un groupe de paramètres de cluster spécifique pour lequel les détails doivent être renvoyés.

Contraintes :

- S'il est fourni, il doit correspondre au nom d'un existant`DBClusterParameterGroup`.

Type : chaîne

Obligatoire : non

Filtres.Filter.N

Ce paramètre n'est actuellement pas pris en charge.

Type : tableau d’objets Filter

Obligatoire : non

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Obligatoire : non

MaxRecords

Nombre maximal d'enregistrements à inclure dans la réponse. S'il existe plus d'enregistrements que la `MaxRecords` valeur spécifiée, un jeton de pagination (marqueur) est inclus dans la réponse afin que les résultats restants puissent être récupérés.

Par défaut : 100

Contraintes : Minimum 20, maximum 100.

Type : entier

Obligatoire : non

Éléments de réponse

Les éléments suivants sont renvoyés par le service.

ClusterParameterGroupsDB D.B. N. ClusterParameterGroup

Liste des groupes de paramètres du cluster.

Type : tableau d’objets DBClusterParameterGroup

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBParameterGroupNotFound

`DBParameterGroupName`ne fait pas référence à un groupe de paramètres existant.

Code d’état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribeDBClusterParameters

Service : Amazon DocumentDB (with MongoDB compatibility)

Renvoie la liste détaillée des paramètres d'un groupe de paramètres de cluster particulier.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterParameterGroupName

Nom d'un groupe de paramètres de cluster spécifique pour lequel les détails des paramètres doivent être renvoyés.

Contraintes :

- S'il est fourni, il doit correspondre au nom d'un existant`DBClusterParameterGroup`.

Type : chaîne

Obligatoire : oui

Filtres.Filter.N

Ce paramètre n'est actuellement pas pris en charge.

Type : tableau d’objets Filter

Obligatoire : non

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Obligatoire : non

MaxRecords

Nombre maximal d'enregistrements à inclure dans la réponse. S'il existe plus d'enregistrements que la `MaxRecords` valeur spécifiée, un jeton de pagination (marqueur) est inclus dans la réponse afin que les résultats restants puissent être récupérés.

Par défaut : 100

Contraintes : Minimum 20, maximum 100.

Type : entier

Obligatoire : non

Source

Valeur indiquée pour renvoyer uniquement les paramètres d'une source spécifique. Les sources du paramètre peuvent être `engine`, `service` ou `customer`.

Type : chaîne

Obligatoire : non

Éléments de réponse

Les éléments suivants sont renvoyés par le service.

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Paramètres.Paramètre.N

Fournit la liste des paramètres du groupe de paramètres de cluster.

Type : tableau d’objets Parameter

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBParameterGroupNotFound

`DBParameterGroupName`ne fait pas référence à un groupe de paramètres existant.

Code d’état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribeDBClusters

Service : Amazon DocumentDB (with MongoDB compatibility)

Renvoie des informations sur les clusters Amazon DocumentDB provisionnés. Cette opération d'API prend en charge la pagination. Pour certaines fonctionnalités de gestion telles que la gestion du cycle de vie des clusters et des instances, Amazon DocumentDB utilise une technologie opérationnelle partagée avec Amazon RDS et Amazon Neptune. Utilisez le paramètre `filterName=engine,Values=docdb` filter pour renvoyer uniquement les clusters Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterIdentifier

L'identifiant de cluster fourni par l'utilisateur. Si ce paramètre est spécifié, les informations provenant uniquement du cluster spécifique sont renvoyées. Ce paramètre n'est pas sensible à la casse.

Contraintes :

- S'il est fourni, il doit correspondre à un existant`DBClusterIdentifier`.

Type : chaîne

Obligatoire : non

Filtres.Filter.N

Filtre qui spécifie un ou plusieurs clusters à décrire.

Filtres pris en charge :

- `db-cluster-id`- Accepte les identifiants de cluster et les Amazon Resource Names (ARN) des clusters. La liste des résultats inclut uniquement des informations sur les clusters identifiés par ces ARN.

Type : tableau d’objets Filter

Obligatoire : non

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Obligatoire : non

MaxRecords

Nombre maximal d'enregistrements à inclure dans la réponse. S'il existe plus d'enregistrements que la `MaxRecords` valeur spécifiée, un jeton de pagination (marqueur) est inclus dans la réponse afin que les résultats restants puissent être récupérés.

Par défaut : 100

Contraintes : Minimum 20, maximum 100.

Type : entier

Obligatoire : non

Éléments de réponse

Les éléments suivants sont renvoyés par le service.

DbClusters.DBCluster.n

Une liste de clusters.

Type : tableau d’objets DBCluster

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterNotFoundFault

`DBClusterIdentifier`ne fait pas référence à un cluster existant.

Code d'état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribeDBClusterSnapshotAttributes

Service : Amazon DocumentDB (with MongoDB compatibility)

Renvoie une liste des noms et des valeurs des attributs d'un instantané de cluster de cluster pour un instantané de cluster de base de données manuel.

Lorsque vous partagez des instantanés avec d'autres `DescribeDBClusterSnapshotAttributes` personnes Comptes AWS, renvoie l'`restore`attribut et une liste des Comptes AWS identifiants autorisés à copier ou à restaurer l'instantané manuel du cluster. S'il `all` est inclus dans la liste des valeurs de l'`restore`attribut, l'instantané manuel du cluster est public et peut être copié ou restauré par tous Comptes AWS.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterSnapshotIdentifier

Identifiant du cliché du cluster dont les attributs doivent être décrits.

Type : chaîne

Obligatoire : oui

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBClusterSnapshotAttributesResult

Informations détaillées sur les attributs associés à un instantané de cluster.

Type : objet DBClusterSnapshotAttributesResult

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`ne fait pas référence à un instantané de cluster existant.

Code d'état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribeDBClusterSnapshots

Service : Amazon DocumentDB (with MongoDB compatibility)

Renvoie des informations sur les instantanés du cluster. Cette opération d'API prend en charge la pagination.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterIdentifier

ID du cluster pour lequel récupérer la liste des instantanés du cluster. Ce paramètre ne peut pas être utilisé avec le `DBClusterSnapshotIdentifier` paramètre. Ce paramètre n'est pas sensible à la casse.

Contraintes :

- S'il est fourni, il doit correspondre à l'identifiant d'un existant`DBCluster`.

Type : chaîne

Obligatoire : non

DBClusterSnapshotIdentifier

Identifiant de capture d'écran spécifique du cluster à décrire. Ce paramètre ne peut pas être utilisé avec le `DBClusterIdentifier` paramètre. Cette valeur est stockée sous la forme d'une chaîne en minuscules.

Contraintes :

- S'il est fourni, il doit correspondre à l'identifiant d'un existant`DBClusterSnapshot`.
- Si cet identifiant est destinée à un instantané automatisé, le paramètre `SnapshotType` doit également être spécifié.

Type : chaîne

Obligatoire : non

Filtres.Filter.N

Ce paramètre n'est actuellement pas pris en charge.

Type : tableau d’objets Filter

Obligatoire : non

IncludePublic

Définissez sur `true` pour inclure les instantanés de cluster manuels qui sont publics et peuvent être copiés ou restaurés par n'importe qui Compte AWS, ou autrement`false`. L’argument par défaut est `false`.

Type : booléen

Obligatoire : non

IncludeShared

Définissez sur `true` pour inclure les instantanés de cluster manuels partagés provenant d'autres Comptes AWS entités autorisées à copier ou à restaurer, etc. Compte AWS `false` L’argument par défaut est `false`.

Type : booléen

Obligatoire : non

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Obligatoire : non

MaxRecords

Nombre maximal d'enregistrements à inclure dans la réponse. S'il existe plus d'enregistrements que la `MaxRecords` valeur spécifiée, un jeton de pagination (marqueur) est inclus dans la réponse afin que les résultats restants puissent être récupérés.

Par défaut : 100

Contraintes : Minimum 20, maximum 100.

Type : entier

Obligatoire : non

SnapshotType

Type de snapshots de cluster à renvoyer. Vous pouvez spécifier l'une des valeurs suivantes :

- `automated`- Renvoie tous les instantanés de cluster qu'Amazon DocumentDB a automatiquement créés pour vous. Compte AWS
- `manual`- Renvoie tous les instantanés de cluster que vous avez créés manuellement pour votre Compte AWS.
- `shared`- Renvoie tous les instantanés de cluster manuels qui ont été partagés avec votre Compte AWS.
- `public`- Renvoie tous les instantanés du cluster marqués comme publics.

Si vous ne spécifiez aucune `SnapshotType` valeur, les instantanés de cluster automatisés et manuels sont renvoyés. Vous pouvez inclure des instantanés de cluster partagés avec ces résultats en définissant le `IncludeShared` paramètre sur. `true` Vous pouvez inclure des instantanés de clusters publics avec ces résultats en définissant le `IncludePublic` paramètre sur. `true`

Les paramètres `IncludeShared` et `IncludePublic` ne s'appliquent pas pour les valeurs `SnapshotType` de `manual` ou `automated`. Le paramètre `IncludePublic` ne s'applique pas lorsque `SnapshotType` est défini sur `shared`. Le paramètre `IncludeShared` ne s'applique pas lorsque `SnapshotType` est défini sur `public`.

Type : chaîne

Obligatoire : non

Éléments de réponse

Les éléments suivants sont renvoyés par le service.

ClusterSnapshotsDB D.B. N. ClusterSnapshot

Fournit une liste des instantanés du cluster.

Type : tableau d'objets DBClusterSnapshot

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`ne fait pas référence à un instantané de cluster existant.

Code d'état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribeDBEngineVersions

Service : Amazon DocumentDB (with MongoDB compatibility)

Renvoie la liste des moteurs disponibles.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBParameterGroupFamily

Nom d'une famille de groupes de paramètres spécifique pour laquelle les informations doivent être renvoyées.

Contraintes :

- S'il est fourni, il doit correspondre à un existant`DBParameterGroupFamily`.

Type : chaîne

Obligatoire : non

DefaultOnly

Indique que seule la version par défaut du moteur spécifié ou de l'association moteur et version majeure est renvoyée.

Type : booléen

Obligatoire : non

Engine

Moteur de base de données à renvoyer.

Type : chaîne

Obligatoire : non

EngineVersion

Version du moteur de base de données à renvoyer.

Exemple : `3.6.0`

Type : chaîne

Obligatoire : non

Filtres.Filter.N

Ce paramètre n'est actuellement pas pris en charge.

Type : tableau d’objets Filter

Obligatoire : non

ListSupportedCharacterSets

Si ce paramètre est spécifié et que le moteur demandé prend en charge le paramètre `CharacterSetName` pour `CreateDBInstance`, la réponse inclut une liste des jeux de caractères pris en charge pour chaque version de moteur.

Type : booléen

Obligatoire : non

ListSupportedTimezones

Si ce paramètre est spécifié et que le moteur demandé prend en charge le paramètre `TimeZone` pour `CreateDBInstance`, la réponse inclut une liste des fuseaux horaires pris en charge pour chaque version de moteur.

Type : booléen

Obligatoire : non

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Obligatoire : non

MaxRecords

Nombre maximal d'enregistrements à inclure dans la réponse. S'il existe plus d'enregistrements que la `MaxRecords` valeur spécifiée, un jeton de pagination (marqueur) est inclus dans la réponse afin que les résultats restants puissent être récupérés.

Par défaut : 100

Contraintes : Minimum 20, maximum 100.

Type : entier

Obligatoire : non

Éléments de réponse

Les éléments suivants sont renvoyés par le service.

EngineVersionsDB D.B. N. EngineVersion

Informations détaillées sur une ou plusieurs versions du moteur.

Type : tableau d'objets DBEngineVersion

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2

- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribeDBInstances

Service : Amazon DocumentDB (with MongoDB compatibility)

Renvoie des informations sur les instances Amazon DocumentDB provisionnées. Cette API prend en charge la pagination.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBInstanceIdentifier

L'identifiant d'instance fourni par l'utilisateur. Si ce paramètre est spécifié, les informations provenant uniquement de l'instance spécifique sont renvoyées. Ce paramètre n'est pas sensible à la casse.

Contraintes :

- S'il est fourni, il doit correspondre à l'identifiant d'un existant`DBInstance`.

Type : chaîne

Obligatoire : non

Filtres.Filter.N

Filtre qui spécifie une ou plusieurs instances à décrire.

Filtres pris en charge :

- `db-cluster-id`- Accepte les identifiants de cluster et les Amazon Resource Names (ARN) des clusters. La liste des résultats inclut uniquement les informations relatives aux instances associées aux clusters identifiés par ces ARN.
- `db-instance-id`- Accepte les identifiants d'instance et les ARN d'instance. La liste des résultats inclut uniquement les informations relatives aux instances identifiées par ces ARN.

Type : tableau d’objets Filter

Obligatoire : non

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Obligatoire : non

MaxRecords

Nombre maximal d'enregistrements à inclure dans la réponse. S'il existe plus d'enregistrements que la `MaxRecords` valeur spécifiée, un jeton de pagination (marqueur) est inclus dans la réponse afin que les résultats restants puissent être récupérés.

Par défaut : 100

Contraintes : Minimum 20, maximum 100.

Type : entier

Obligatoire : non

Éléments de réponse

Les éléments suivants sont renvoyés par le service.

DBInstances.DBInstance.n

Informations détaillées sur une ou plusieurs instances.

Type : tableau d’objets DBInstance

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBInstanceNotFound

`DBInstanceIdentifier`ne fait pas référence à une instance existante.

Code d'état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribeDBSubnetGroups

Service : Amazon DocumentDB (with MongoDB compatibility)

Renvoie une liste des descriptions de `DBSubnetGroup`. Si a `DBSubnetGroupName` est spécifié, la liste contiendra uniquement les descriptions du spécifié`DBSubnetGroup`.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBSubnetGroupName

Nom du groupe de sous-réseaux pour lequel les informations doivent être renvoyées.

Type : chaîne

Obligatoire : non

Filtres.Filter.N

Ce paramètre n'est actuellement pas pris en charge.

Type : tableau d’objets Filter

Obligatoire : non

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Obligatoire : non

MaxRecords

Nombre maximal d'enregistrements à inclure dans la réponse. S'il existe plus d'enregistrements que la `MaxRecords` valeur spécifiée, un jeton de pagination (marqueur) est inclus dans la réponse afin que les résultats restants puissent être récupérés.

Par défaut : 100

Contraintes : Minimum 20, maximum 100.

Type : entier

Obligatoire : non

Éléments de réponse

Les éléments suivants sont renvoyés par le service.

SubnetGroupsDB D.B. N. SubnetGroup

Informations détaillées sur un ou plusieurs groupes de sous-réseaux.

Type : tableau d'objets DBSubnetGroup

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`ne fait pas référence à un groupe de sous-réseaux existant.

Code d'état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS

- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribeEngineDefaultClusterParameters

Service : Amazon DocumentDB (with MongoDB compatibility)

Renvoie les informations sur les paramètres de moteur et de système par défaut du moteur de base de données du cluster.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBParameterGroupFamily

Nom de la famille de groupes de paramètres du cluster pour laquelle les informations sur les paramètres du moteur doivent être renvoyées.

Type : chaîne

Obligatoire : oui

Filtres.Filter.N

Ce paramètre n'est actuellement pas pris en charge.

Type : tableau d’objets Filter

Obligatoire : non

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Obligatoire : non

MaxRecords

Nombre maximal d'enregistrements à inclure dans la réponse. S'il existe plus d'enregistrements que la `MaxRecords` valeur spécifiée, un jeton de pagination (marqueur) est inclus dans la réponse afin que les résultats restants puissent être récupérés.

Par défaut : 100

Contraintes : Minimum 20, maximum 100.

Type : entier

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

EngineDefaults

Contient le résultat d'une invocation réussie de l'`DescribeEngineDefaultClusterParameters`opération.

Type : objet EngineDefaults

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribeEventCategories

Service : Amazon DocumentDB (with MongoDB compatibility)

Affiche une liste des catégories de tous les types de sources de l'événement ou, si la valeur est spécifiée, d'un type de source donné.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

Filtres.Filter.N

Ce paramètre n'est actuellement pas pris en charge.

Type : tableau d’objets Filter

Obligatoire : non

SourceType

Type de source qui génère les événements.

Valeurs valides: `db-instance`, `db-parameter-group`, `db-security-group`

Type : chaîne

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

EventCategoriesMapList. EventCategoriesMapN.

Liste des cartes des catégories d'événements.

Type : tableau d’objets EventCategoriesMap

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribeEvents

Service : Amazon DocumentDB (with MongoDB compatibility)

Renvoie les événements relatifs aux instances, aux groupes de sécurité, aux instantanés et aux groupes de paramètres de base de données des 14 derniers jours. Vous pouvez obtenir des événements spécifiques à une instance de base de données, à un groupe de sécurité, à un instantané ou à un groupe de paramètres en fournissant le nom en tant que paramètre. Par défaut, les événements de l'heure passée sont renvoyés.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

Duration

Nombre de minutes pour lesquelles récupérer les événements.

Par défaut : 60

Type : entier

Obligatoire : non

EndTime

Fin de l'intervalle de temps pour lequel récupérer les événements au format ISO 8601.

Exemple : 2009-07-08T18:00Z

Type : Timestamp

Obligatoire : non

EventCategories. EventCategoryN.

Liste des catégories d'événements qui déclenchent des notifications pour un abonnement aux notifications d'événements.

Type : tableau de chaînes

Obligatoire : non

Filtres.Filter.N

Ce paramètre n'est actuellement pas pris en charge.

Type : tableau d’objets Filter

Obligatoire : non

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Obligatoire : non

MaxRecords

Nombre maximal d'enregistrements à inclure dans la réponse. S'il existe plus d'enregistrements que la `MaxRecords` valeur spécifiée, un jeton de pagination (marqueur) est inclus dans la réponse afin que les résultats restants puissent être récupérés.

Par défaut : 100

Contraintes : Minimum 20, maximum 100.

Type : entier

Obligatoire : non

SourceIdentifier

Identifiant de la source de l'événement pour laquelle les événements sont renvoyés. Si la valeur n'est pas spécifiée, toutes les sources sont incluses dans la réponse.

Contraintes :

- S'il `SourceIdentifier` est fourni, `SourceType` il doit également être fourni.
- Si le type de source est`DBInstance`, un `DBInstanceIdentifier` doit être fourni.
- Si le type de source est`DBSecurityGroup`, un `DBSecurityGroupName` doit être fourni.
- Si le type de source est`DBParameterGroup`, un `DBParameterGroupName` doit être fourni.

- Si le type de source est`DBSnapshot`, un `DBSnapshotIdentifier` doit être fourni.
- Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

Type : chaîne

Obligatoire : non

SourceType

Source de l'événement pour laquelle récupérer les événements. Si aucune valeur n'est spécifiée, tous les événements sont renvoyés.

Type : chaîne

Valeurs valides : `db-instance | db-parameter-group | db-security-group | db-snapshot | db-cluster | db-cluster-snapshot`

Obligatoire : non

StartTime

Début de l'intervalle de temps pour lequel récupérer les événements au format ISO 8601.

Exemple : 2009-07-08T18:00Z

Type : Timestamp

Obligatoire : non

Éléments de réponse

Les éléments suivants sont renvoyés par le service.

Événements.Événement.N

Informations détaillées sur un ou plusieurs événements.

Type : tableau d’objets Event

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribeEventSubscriptions

Service : Amazon DocumentDB (with MongoDB compatibility)

Répertorie toutes les descriptions d'abonnements d'un compte client. La description d'un abonnement inclut `SubscriptionNameSNSTopicARN`,`CustomerID`, `SourceTypeSourceID`,`CreationTime`, et`Status`.

Si vous spécifiez un`SubscriptionName`, répertorie la description de cet abonnement.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

Filtres.Filter.N

Ce paramètre n'est actuellement pas pris en charge.

Type : tableau d’objets Filter

Obligatoire : non

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Obligatoire : non

MaxRecords

Nombre maximal d'enregistrements à inclure dans la réponse. S'il existe plus d'enregistrements que la `MaxRecords` valeur spécifiée, un jeton de pagination (marqueur) est inclus dans la réponse afin que les résultats restants puissent être récupérés.

Par défaut : 100

Contraintes : Minimum 20, maximum 100.

Type : entier

Obligatoire : non

SubscriptionName

Nom de l'abonnement aux notifications d'événements Amazon DocumentDB que vous souhaitez décrire.

Type : chaîne

Obligatoire : non

Éléments de réponse

Les éléments suivants sont renvoyés par le service.

EventSubscriptionsList. EventSubscriptionN.

Liste des abonnements aux événements.

Type : tableau d’objets EventSubscription

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

SubscriptionNotFound

Le nom de l'abonnement n'existe pas.

Code d’état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribeGlobalClusters

Service : Amazon DocumentDB (with MongoDB compatibility)

Renvoie des informations sur les clusters globaux Amazon DocumentDB. Cette API prend en charge la pagination.

> **Note**
>
> Cette action s'applique uniquement aux clusters Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

Filtres.Filter.N

Filtre qui spécifie un ou plusieurs clusters de bases de données globaux à décrire.

Filtres pris en charge : `db-cluster-id` accepte les identifiants de cluster et les Amazon Resource Names (ARN) des clusters. La liste des résultats inclura uniquement des informations sur les clusters identifiés par ces ARN.

Type : tableau d'objets Filter

Obligatoire : non

GlobalClusterIdentifier

L'identifiant de cluster fourni par l'utilisateur. Si ce paramètre est spécifié, les informations provenant uniquement du cluster spécifique sont renvoyées. Ce paramètre n'est pas sensible à la casse.

Type : chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 255.

Modèle : `[A-Za-z][0-9A-Za-z-:._]*`

Obligatoire : non

Marker

Jeton de pagination facultatif fourni par une demande `DescribeGlobalClusters` précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Obligatoire : non

MaxRecords

Nombre maximal d'enregistrements à inclure dans la réponse. S'il existe plus d'enregistrements que la `MaxRecords` valeur spécifiée, un jeton de pagination appelé marqueur est inclus dans la réponse afin que vous puissiez récupérer les résultats restants.

Type : entier

Obligatoire : non

Éléments de réponse

Les éléments suivants sont renvoyés par le service.

GlobalClusters. GlobalClusterMemberN.

Type : tableau d’objets GlobalCluster

Marker

Type : chaîne

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

GlobalClusterNotFoundFault

`GlobalClusterIdentifier`Cela ne fait pas référence à un cluster mondial existant.

Code d’état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribeOrderableDBInstanceOptions

Service : Amazon DocumentDB (with MongoDB compatibility)

Renvoie une liste d'options d'instance pouvant être commandées pour le moteur spécifié.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

Engine

Nom du moteur pour lequel les options d'instance doivent être récupérées.

Type : chaîne

Obligatoire : oui

DBInstanceClass

La valeur du filtre de classe d'instance. Spécifiez ce paramètre pour afficher uniquement les offres disponibles correspondant à la classe d'instance spécifiée.

Type : chaîne

Obligatoire : non

EngineVersion

Valeur de filtre de la version de moteur. Spécifiez ce paramètre pour afficher uniquement les offres disponibles correspondant à la version du moteur spécifiée.

Type : chaîne

Obligatoire : non

Filtres.Filter.N

Ce paramètre n'est actuellement pas pris en charge.

Type : tableau d’objets Filter

Obligatoire : non

LicenseModel

Valeur de filtre du modèle de licence. Spécifiez ce paramètre pour afficher uniquement les offres disponibles correspondant au modèle de licence spécifié.

Type : chaîne

Obligatoire : non

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Obligatoire : non

MaxRecords

Nombre maximal d'enregistrements à inclure dans la réponse. S'il existe plus d'enregistrements que la `MaxRecords` valeur spécifiée, un jeton de pagination (marqueur) est inclus dans la réponse afin que les résultats restants puissent être récupérés.

Par défaut : 100

Contraintes : Minimum 20, maximum 100.

Type : entier

Obligatoire : non

Vpc

La valeur du filtre de cloud privé virtuel (VPC). Spécifiez ce paramètre pour afficher uniquement les offres VPC ou non VPC disponibles.

Type : booléen

Obligatoire : non

Éléments de réponse

Les éléments suivants sont renvoyés par le service.

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

OrderableDB .OrderableDB N. InstanceOptions InstanceOption

Les options disponibles pour une instance commandable particulière.

Type : tableau d'objets OrderableDBInstanceOption

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DescribePendingMaintenanceActions

Service : Amazon DocumentDB (with MongoDB compatibility)

Renvoie une liste de ressources (par exemple, des instances) pour lesquelles au moins une action de maintenance est en attente.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

Filtres.Filter.N

Filtre qui spécifie une ou plusieurs ressources pour lesquelles renvoyer des actions de maintenance en attente.

Filtres pris en charge :

- `db-cluster-id`- Accepte les identifiants de cluster et les Amazon Resource Names (ARN) des clusters. La liste des résultats inclut uniquement les actions de maintenance en attente pour les clusters identifiés par ces ARN.
- `db-instance-id`- Accepte les identifiants d'instance et les ARN d'instance. La liste des résultats inclut uniquement les actions de maintenance en attente pour les instances de base de données identifiées par ces ARN.

Type : tableau d’objets Filter

Obligatoire : non

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Obligatoire : non

MaxRecords

Nombre maximal d'enregistrements à inclure dans la réponse. S'il existe plus d'enregistrements que la `MaxRecords` valeur spécifiée, un jeton de pagination (marqueur) est inclus dans la réponse afin que les résultats restants puissent être récupérés.

Par défaut : 100

Contraintes : Minimum 20, maximum 100.

Type : entier

Obligatoire : non

ResourceIdentifier

ARN d'une ressource pour laquelle renvoyer des actions de maintenance en attente.

Type : chaîne

Obligatoire : non

Éléments de réponse

Les éléments suivants sont renvoyés par le service.

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

PendingMaintenanceActions. ResourcePendingMaintenanceActionsN.

Les actions de maintenance à appliquer.

Type : tableau d’objets ResourcePendingMaintenanceActions

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

ResourceNotFoundFault

L'ID de ressource spécifiée est introuvable.

Code d’état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

FailoverDBCluster

Service : Amazon DocumentDB (with MongoDB compatibility)

Force le basculement d'un cluster.

En cas de basculement d'un cluster, l'une des répliques Amazon DocumentDB (instances en lecture seule) du cluster devient l'instance principale (le rédacteur du cluster).

En cas de défaillance de l'instance principale, Amazon DocumentDB bascule automatiquement vers une réplique Amazon DocumentDB, le cas échéant. Vous pouvez forcer un basculement lorsque vous souhaitez simuler un échec d'une instance principale à des fins de test.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterIdentifier

Identifiant de cluster pour lequel forcer un basculement. Ce paramètre n'est pas sensible à la casse.

Contraintes :

- Doit correspondre à l'identifiant d'un `DBCluster` existant.

Type : chaîne

Obligatoire : non

TargetDBInstanceIdentifier

Nom de l'instance à promouvoir en instance principale.

Vous devez spécifier l'identifiant d'instance pour une réplique Amazon DocumentDB dans le cluster. Par exemple, `mydbcluster-replica1`.

Type : chaîne

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBCluster

Informations détaillées sur un cluster.

Type : objet DBCluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterNotFoundFault

`DBClusterIdentifier`ne fait pas référence à un cluster existant.

Code d'état HTTP : 404

InvalidDBClusterStateFault

L'état du cluster n'est pas valide.

Code d'état HTTP : 400

InvalidDBInstanceState

L'instance spécifiée n'est pas disponible.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3

- AWS SDK pour Python
- AWS SDK pour Ruby V3

ListTagsForResource

Service : Amazon DocumentDB (with MongoDB compatibility)

Répertorie toutes les balises d'une ressource Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

ResourceName

La ressource Amazon DocumentDB avec les balises à répertorier. Cette valeur est un Amazon Resource Name (ARN).

Type : chaîne

Obligatoire : oui

Filtres.Filter.N

Ce paramètre n'est actuellement pas pris en charge.

Type : tableau d’objets Filter

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

TagList.Tag N

Liste d'un ou de plusieurs tags.

Type : tableau d’objets Tag

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterNotFoundFault

`DBClusterIdentifier`ne fait pas référence à un cluster existant.

Code d'état HTTP : 404

DBInstanceNotFound

`DBInstanceIdentifier`ne fait pas référence à une instance existante.

Code d'état HTTP : 404

DBSnapshotNotFound

`DBSnapshotIdentifier`ne fait pas référence à un instantané existant.

Code d'état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

ModifyDBCluster

Service : Amazon DocumentDB (with MongoDB compatibility)

Modifie un paramètre pour un cluster Amazon DocumentDB. Vous pouvez modifier un ou plusieurs paramètres de configuration de base de données en spécifiant ces paramètres et les nouvelles valeurs dans la demande.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterIdentifier

Identifiant du cluster en cours de modification. Ce paramètre n'est pas sensible à la casse.

Contraintes :

- Doit correspondre à l'identifiant d'un `DBCluster` existant.

Type : chaîne

Obligatoire : oui

AllowMajorVersionUpgrade

Une valeur qui indique que les mises à niveau de version majeures sont autorisées.

Contraintes : vous devez autoriser les mises à niveau des versions majeures lorsque vous spécifiez une valeur pour le `EngineVersion` paramètre qui est une version majeure différente de la version actuelle du cluster de base de données.

Type : booléen

Obligatoire : non

ApplyImmediately

Valeur qui indique si les modifications apportées à cette demande et aux modifications en attente sont appliquées de manière asynchrone dès que possible, quel que soit le `PreferredMaintenanceWindow` paramètre du cluster. Si ce paramètre est défini sur`false`, les modifications apportées au cluster sont appliquées lors de la fenêtre de maintenance suivante.

Le `ApplyImmediately` paramètre affecte uniquement les `MasterUserPassword` valeurs `NewDBClusterIdentifier` et. Si vous définissez la valeur de ce paramètre sur`false`, les modifications apportées aux `MasterUserPassword` valeurs `NewDBClusterIdentifier` et sont appliquées lors de la fenêtre de maintenance suivante. Toutes les autres modifications sont appliquées immédiatement, quelle que soit la valeur du paramètre `ApplyImmediately`.

Par défaut : `false`

Type : booléen

Obligatoire : non

BackupRetentionPeriod

Nombre de jours de conservation des sauvegardes automatiques. Vous devez spécifier une valeur minimale de 1.

Par défaut : 1

Contraintes :

- Doit être une valeur comprise entre 1 et 35.

Type : entier

Obligatoire : non

CloudwatchLogsExportConfiguration

Le paramètre de configuration des types de journaux à activer pour l'exportation vers Amazon CloudWatch Logs pour une instance ou un cluster spécifique. Les `DisableLogTypes` tableaux `EnableLogTypes` et déterminent quels journaux sont exportés (ou non exportés) vers CloudWatch Logs.

Type : objet CloudwatchLogsExportConfiguration

Obligatoire : non

DBClusterParameterGroupName

Le nom du groupe de paramètres de cluster à utiliser pour le cluster.

Type : chaîne

Obligatoire : non

DeletionProtection

Spécifie si ce cluster peut être supprimé. Si cette option `DeletionProtection` est activée, le cluster ne peut pas être supprimé sauf s'il `DeletionProtection` est modifié et désactivé. `DeletionProtection`protège les clusters contre la suppression accidentelle.

Type : booléen

Obligatoire : non

EngineVersion

Numéro de version du moteur de base de données vers lequel vous souhaitez effectuer la mise à niveau. La modification de ce paramètre entraîne une interruption. La modification sera appliquée pendant la fenêtre de maintenance suivante, sauf si `ApplyImmediately` est activé.

Pour répertorier toutes les versions de moteur disponibles pour Amazon DocumentDB, utilisez la commande suivante :

```
aws docdb describe-db-engine-versions --engine docdb --query
"DBEngineVersions[].EngineVersion"
```

Type : chaîne

Obligatoire : non

MasterUserPassword

Mot de passe de l'utilisateur principal de la base de données. Ce mot de passe peut contenir tout caractère ASCII imprimable à l'exception de la barre oblique (/), des guillemets doubles (") ou du symbole arobase (@).

Contraintes : doit comporter entre 8 et 100 caractères.

Type : chaîne

Obligatoire : non

NewDBClusterIdentifier

Le nouvel identificateur de cluster pour le cluster lors du changement de nom d'un cluster. Cette valeur est stockée sous la forme d'une chaîne en minuscules.

Contraintes :

- Doit contenir entre 1 et 63 lettres, chiffres ou traits d'union.
- Le premier caractère doit être une lettre.
- Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

Exemple : `my-cluster2`

Type : chaîne

Obligatoire : non

Port

Numéro de port au niveau duquel le cluster accepte des connexions.

Contraintes : Doit être une valeur comprise entre `1150` et`65535`.

Par défaut : le même port que le cluster d'origine.

Type : entier

Obligatoire : non

PreferredBackupWindow

Plage de temps quotidienne au cours de laquelle les sauvegardes automatiques sont créées si cette fonctionnalité est activée via le paramètre `BackupRetentionPeriod`.

La valeur par défaut est une fenêtre de 30 minutes sélectionnée au hasard dans un intervalle de 8 heures pour chacune d'entre elles. Région AWS

Contraintes :

- Doit être au format `hh24:mi-hh24:mi`.
- Doit être exprimée en heure UTC (Universal Coordinated Time).
- Ne doit pas être en conflit avec la fenêtre de maintenance préférée.
- Doit être de 30 minutes minimum.

Type : chaîne

Obligatoire : non

PreferredMaintenanceWindow

Intervalle de temps hebdomadaire, au format Universal Coordinated Time (UTC), pendant lequel a lieu la maintenance du système.

Format : `ddd:hh24:mi-ddd:hh24:mi`

La valeur par défaut est une fenêtre de 30 minutes sélectionnée au hasard dans un intervalle de 8 heures pour chacune d'elles Région AWS, survenant un jour aléatoire de la semaine.

Jours valides : Mon, Tue, Wed, Thu, Fri, Sat, Sun

Contraintes : fenêtre minimale de 30 minutes.

Type : chaîne

Obligatoire : non

StorageType

Type de stockage à associer au cluster de base de données.

Pour plus d'informations sur les types de stockage pour les clusters Amazon DocumentDB, consultez la section Configurations de stockage des clusters dans le manuel Amazon DocumentDB Developer Guide.

Valeurs valides pour le type de stockage - `standard | iopt1`

La valeur par défaut est `standard`

Type : chaîne

Obligatoire : non

VpcSecurityGroupIds. VpcSecurityGroupIdN.

Liste des groupes de sécurité du cloud privé virtuel (VPC) auxquels le cluster appartiendra.

Type : tableau de chaînes

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBCluster

Informations détaillées sur un cluster.

Type : objet DBCluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterAlreadyExistsFault

Vous avez déjà un cluster avec l'identifiant indiqué.

Code d’état HTTP : 400

DBClusterNotFoundFault

`DBClusterIdentifier`ne fait pas référence à un cluster existant.

Code d’état HTTP : 404

DBClusterParameterGroupNotFound

`DBClusterParameterGroupName`ne fait pas référence à un groupe de paramètres de cluster existant.

Code d’état HTTP : 404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`ne fait pas référence à un groupe de sous-réseaux existant.

Code d’état HTTP : 404

InvalidDBClusterStateFault

Le cluster n'est pas dans un état valide.

Code d’état HTTP : 400

InvalidDBInstanceState

L'instance spécifiée n'est pas disponible.

Code d’état HTTP : 400

InvalidDBSecurityGroupState

L'état du groupe de sécurité n'autorise pas la suppression.

Code d’état HTTP : 400

InvalidDBSubnetGroupStateFault

Le groupe de sous-réseaux ne peut pas être supprimé car il est en cours d'utilisation.

Code d’état HTTP : 400

InvalidSubnet

Le sous-réseau demandé n'est pas valide ou plusieurs sous-réseaux ont été demandés mais ils ne se trouvent pas tous dans un cloud privé virtuel (VPC) commun.

Code d’état HTTP : 400

InvalidVPCNetworkStateFault

Le groupe de sous-réseaux ne couvre pas toutes les zones de disponibilité après sa création en raison des modifications apportées.

Code d’état HTTP : 400

StorageQuotaExceeded

La demande vous obligerait à dépasser la quantité de stockage autorisée disponible sur toutes les instances.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python

- AWS SDK pour Ruby V3

ModifyDBClusterParameterGroup

Service : Amazon DocumentDB (with MongoDB compatibility)

Modifie les paramètres d'un groupe de paramètres de cluster. Pour modifier plusieurs paramètres, soumettez une liste des éléments suivants : `ParameterName`, `ParameterValue` et `ApplyMethod`. 20 paramètres maximum peuvent être modifiés dans une même demande.

> **Note**
>
> Les modifications apportées aux paramètres dynamiques sont appliquées immédiatement. Les modifications apportées aux paramètres statiques nécessitent un redémarrage ou une fenêtre de maintenance avant que la modification ne prenne effet.

> **Important**
>
> Après avoir créé un groupe de paramètres de cluster, vous devez patienter au moins 5 minutes avant de créer votre premier cluster devant utiliser ce groupe de paramètres de cluster comme groupe de paramètres par défaut. Cela permet à Amazon DocumentDB de terminer complètement l'action de création avant que le groupe de paramètres ne soit utilisé par défaut pour un nouveau cluster. Cette étape est particulièrement importante pour les paramètres qui sont essentiels lors de la création de la base de données par défaut d'un cluster, tels que le jeu de caractères de la base de données par défaut définie par le paramètre `character_set_database`.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterParameterGroupName

Nom du groupe de paramètres de cluster à modifier.

Type : chaîne

Obligatoire : oui

Paramètres.Paramètre.N

Liste des paramètres du groupe de paramètres du cluster à modifier.

Type : tableau d'objets Parameter

Obligatoire : oui

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBClusterParameterGroupName

Nom d'un groupe de paramètres de cluster.

Contraintes :

- Doit contenir de 1 à 255 lettres ou chiffres.
- Le premier caractère doit être une lettre.
- Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

> **Note**
>
> Cette valeur est stockée sous la forme d'une chaîne en minuscules.

Type : chaîne

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBParameterGroupNotFound

`DBParameterGroupName`ne fait pas référence à un groupe de paramètres existant.

Code d'état HTTP : 404

InvalidDBParameterGroupState

Le groupe de paramètres est en cours d'utilisation ou son état n'est pas valide. Si vous essayez de supprimer le groupe de paramètres, vous ne pouvez pas le supprimer lorsque le groupe de paramètres est dans cet état.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

ModifyDBClusterSnapshotAttribute

Service : Amazon DocumentDB (with MongoDB compatibility)

Ajoute un attribut et des valeurs ou supprime un attribut et des valeurs d'un instantané de cluster manuel.

Pour partager un instantané de cluster manuel avec d'autres personnes Comptes AWS`AttributeName`, spécifiez `restore` comme `ValuesToAdd` paramètre et utilisez le paramètre pour ajouter une liste des Comptes AWS identifiants autorisés à restaurer l'instantané de cluster manuel. Utilisez cette valeur `all` pour rendre public le cliché manuel du cluster, ce qui signifie qu'il peut être copié ou restauré par tous Comptes AWS. N'ajoutez aucune `all` valeur aux instantanés de cluster manuels contenant des informations privées que vous ne souhaitez pas voir accessibles à tous Comptes AWS. Si un instantané de cluster manuel est chiffré, il peut être partagé, mais uniquement en spécifiant une liste d' Compte AWS identifiants autorisés pour le `ValuesToAdd` paramètre. Dans ce cas, vous ne pouvez pas utiliser `all` comme une valeur pour ce paramètre.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

AttributeName

Nom de l'attribut de capture d'écran du cluster à modifier.

Pour gérer l'autorisation permettant Comptes AWS à d'autres personnes de copier ou de restaurer un instantané de cluster manuel, définissez cette valeur sur`restore`.

Type : chaîne

Obligatoire : oui

DBClusterSnapshotIdentifier

Identifiant du cliché du cluster dont les attributs doivent être modifiés.

Type : chaîne

Obligatoire : oui

ValuesToAdd. AttributeValueN.

Liste des attributs de capture d'écran du cluster à ajouter à l'attribut spécifié par`AttributeName`.

Pour autoriser d'autres Comptes AWS utilisateurs à copier ou à restaurer un instantané de cluster manuel, configurez cette liste pour inclure un ou plusieurs Compte AWS identifiants. Pour que l'instantané manuel du cluster puisse être restauré par n'importe qui Compte AWS, définissez-le sur`all`. N'ajoutez aucune `all` valeur aux instantanés de cluster manuels contenant des informations privées que vous ne souhaitez pas rendre accessibles à tous Comptes AWS.

Type : tableau de chaînes

Obligatoire : non

ValuesToRemove. AttributeValueN.

Liste des attributs de capture d'écran du cluster à supprimer de l'attribut spécifié par`AttributeName`.

Pour supprimer l'autorisation permettant Comptes AWS à d'autres utilisateurs de copier ou de restaurer un instantané de cluster manuel, configurez cette liste pour inclure un ou plusieurs Compte AWS identifiants. Pour supprimer l'autorisation permettant Compte AWS à quiconque de copier ou de restaurer l'instantané du cluster, définissez-le sur`all`. Si vous le spécifiez`all`, un Compte AWS utilisateur dont l'ID de compte est explicitement ajouté à l'`restore`attribut peut toujours copier ou restaurer un instantané de cluster manuel.

Type : tableau de chaînes

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBClusterSnapshotAttributesResult

Informations détaillées sur les attributs associés à un instantané de cluster.

Type : objet DBClusterSnapshotAttributesResult

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`ne fait pas référence à un instantané de cluster existant.

Code d'état HTTP : 404

InvalidDBClusterSnapshotStateFault

La valeur fournie n'est pas un état de capture d'écran de cluster valide.

Code d'état HTTP : 400

SharedSnapshotQuotaExceeded

Vous avez dépassé le nombre maximal de comptes avec lesquels vous pouvez partager un instantané de base de données manuel.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

ModifyDBInstance

Service : Amazon DocumentDB (with MongoDB compatibility)

Modifie les paramètres d'une instance. Vous pouvez modifier un ou plusieurs paramètres de configuration de base de données en spécifiant ces paramètres et les nouvelles valeurs dans la demande.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBInstanceIdentifier

Identifiant de l’instance. Cette valeur est stockée sous la forme d'une chaîne en minuscules.

Contraintes :

- Doit correspondre à l'identifiant d'un `DBInstance` existant.

Type : chaîne

Obligatoire : oui

ApplyImmediately

Spécifie si les modifications de cette demande et les modifications en attente sont appliquées de manière asynchrone dès que possible, quel que soit le `PreferredMaintenanceWindow` paramètre de l'instance.

Si ce paramètre est défini sur`false`, les modifications apportées à l'instance sont appliquées lors de la fenêtre de maintenance suivante. Certaines modifications de paramètres peuvent provoquer une panne et sont appliquées lors du prochain redémarrage.

Par défaut : `false`

Type : booléen

Obligatoire : non

AutoMinorVersionUpgrade

Ce paramètre ne s'applique pas à Amazon DocumentDB. Amazon DocumentDB n'effectue pas de mises à niveau mineures de version, quelle que soit la valeur définie.

Type : booléen

Obligatoire : non

CACertificateIdentifier

Indique le certificat qui doit être associé à l'instance.

Type : chaîne

Obligatoire : non

CertificateRotationRestart

Spécifie si l'instance de base de données est redémarrée lorsque vous faites pivoter votre certificat SSL/TLS.

Par défaut, l'instance de base de données est redémarrée lorsque vous faites pivoter votre certificat SSL/TLS. Le certificat n'est pas mis à jour tant que l'instance de base de données n'est pas redémarrée.

> ⚠ Important
>
> Définissez ce paramètre uniquement si vous n'utilisez pas le protocole SSL/TLS pour vous connecter à l'instance de base de données.

Si vous utilisez le protocole SSL/TLS pour vous connecter à l'instance de base de données, consultez la section Mise à jour de vos certificats Amazon DocumentDB TLS et chiffrement des données en transit dans le manuel Amazon DocumentDB Developer Guide.

Type : booléen

Obligatoire : non

CopyTagsToSnapshot

Une valeur qui indique s'il faut copier toutes les balises de l'instance de base de données vers des instantanés de l'instance de base de données. Par défaut, les balises ne sont pas copiées.

Type : booléen

Obligatoire : non

DBInstanceClass

La nouvelle capacité de calcul et de mémoire de l'instance ; par exemple,`db.r5.large`. Les classes d'instance ne sont pas toutes disponibles Régions AWS.

Si vous modifiez la classe d'instance, une panne se produit lors de la modification. La modification est appliquée pendant la fenêtre de maintenance suivante, sauf si `ApplyImmediately` a été spécifié `true` pour cette demande.

Par défaut : Utilise le paramètre existant.

Type : chaîne

Obligatoire : non

EnablePerformanceInsights

Une valeur qui indique s'il convient d'activer Performance Insights pour l'instance de base de données. Pour plus d'informations, voir Utilisation d'Amazon Performance Insights.

Type : booléen

Obligatoire : non

NewDBInstanceIdentifier

Le nouvel identifiant d'instance pour l'instance lorsque vous renommez une instance. Lorsque vous modifiez l'identifiant de l'instance, un redémarrage de l'instance a lieu immédiatement si vous le définissez `Apply Immediately` sur`true`. Cela se produit lors de la fenêtre de maintenance suivante si vous définissez `Apply Immediately` sur`false`. Cette valeur est stockée sous la forme d'une chaîne en minuscules.

Contraintes :

- Doit contenir entre 1 et 63 lettres, chiffres ou traits d'union.
- Le premier caractère doit être une lettre.
- Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

Exemple : `mydbinstance`

Type : chaîne

Obligatoire : non

PerformanceInsightsKMSKeyId

Identifiant AWS KMS clé pour le chiffrement des données Performance Insights.

L'identifiant de AWS KMS clé est l'ARN de la clé, l'ID de clé, l'alias ARN ou le nom d'alias de la clé KMS.

Si vous ne spécifiez aucune valeur pour PerformanceInsights KMSKeyId, Amazon DocumentDB utilise votre clé KMS par défaut. Il existe une clé KMS par défaut pour votre compte Amazon Web Services. Votre compte Amazon Web Services possède une clé KMS par défaut différente pour chaque région Amazon Web Services.

Type : chaîne

Obligatoire : non

PreferredMaintenanceWindow

Intervalle de temps hebdomadaire (au format UTC) pendant lequel se produit la maintenance du système qui peut entraîner une interruption. La modification de ce paramètre n'entraîne pas de panne, sauf dans le cas suivant, et la modification est appliquée de manière asynchrone dès que possible. Si des actions en attente entraînent un redémarrage et que la fenêtre de maintenance est modifiée pour inclure l'heure actuelle, la modification de ce paramètre entraîne le redémarrage de l'instance. Si vous déplacez cette fenêtre vers l'heure actuelle, il doit s'écouler au moins 30 minutes entre l'heure actuelle et la fin de la fenêtre pour garantir que les modifications en attente sont appliquées.

Par défaut : Utilise le paramètre existant.

Format : `ddd:hh24:mi-ddd:hh24:mi`

Jours valides : Mon, Tue, Wed, Thu, Fri, Sat, Sun

Contraintes : Doit durer au moins 30 minutes.

Type : chaîne

Obligatoire : non

PromotionTier

Valeur qui indique l'ordre dans lequel une réplique Amazon DocumentDB est promue vers l'instance principale après une défaillance de l'instance principale existante.

Valeur par défaut : 1

Valeurs valides : 0 à 15

Type : entier

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBInstance

Informations détaillées sur une instance.

Type : objet DBInstance

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

AuthorizationNotFound

L'adresse IP CIDR ou le groupe de sécurité Amazon EC2 spécifié n'est pas autorisé pour le groupe de sécurité spécifié.

Amazon DocumentDB peut également ne pas être autorisé à effectuer les actions nécessaires en votre nom à l'aide d'IAM.

Code d'état HTTP : 404

CertificateNotFound

`CertificateIdentifier`ne fait pas référence à un certificat existant.

Code d'état HTTP : 404

DBInstanceAlreadyExists

Vous avez déjà une instance avec l'identifiant indiqué.

Code d'état HTTP : 400

DBInstanceNotFound

`DBInstanceIdentifier`ne fait pas référence à une instance existante.

Code d'état HTTP : 404

DBParameterGroupNotFound

`DBParameterGroupName`ne fait pas référence à un groupe de paramètres existant.

Code d'état HTTP : 404

DBSecurityGroupNotFound

`DBSecurityGroupName`ne fait pas référence à un groupe de sécurité existant.

Code d'état HTTP : 404

DBUpgradeDependencyFailure

La mise à niveau a échoué car une ressource dont dépend ne peut pas être modifiée.

Code d'état HTTP : 400

InsufficientDBInstanceCapacity

La classe d'instance spécifiée n'est pas disponible dans la zone de disponibilité spécifiée.

Code d'état HTTP : 400

InvalidDBInstanceState

L'instance spécifiée n'est pas dans l'état disponible.

Code d'état HTTP : 400

InvalidDBSecurityGroupState

L'état du groupe de sécurité n'autorise pas la suppression.

Code d'état HTTP : 400

InvalidVPCNetworkStateFault

Le groupe de sous-réseaux ne couvre pas toutes les zones de disponibilité après sa création en raison des modifications apportées.

Code d'état HTTP : 400

StorageQuotaExceeded

La demande vous obligerait à dépasser la quantité de stockage autorisée disponible sur toutes les instances.

Code d'état HTTP : 400

StorageTypeNotSupported

Le stockage du paramètre spécifié ne `StorageType` peut pas être associé à l'instance de base de données.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

ModifyDBSubnetGroup

Service : Amazon DocumentDB (with MongoDB compatibility)

Modifie un groupe de sous-réseaux existant. Les groupes de sous-réseaux doivent contenir au moins un sous-réseau dans au moins deux zones de disponibilité du. Région AWS

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBSubnetGroupName

Nom du groupe de sous-réseaux. Cette valeur est stockée sous la forme d'une chaîne en minuscules. Vous ne pouvez pas modifier le groupe de sous-réseaux par défaut.

Contraintes : doit correspondre au nom d'un `DBSubnetGroup` existant. Impossible de conserver le nom par défaut.

Exemple : `mySubnetgroup`

Type : chaîne

Obligatoire : oui

SubnetIds. SubnetIdentifierN.

ID de sous-réseau Amazon EC2 du groupe de sous-réseaux.

Type : tableau de chaînes

Obligatoire : oui

DBSubnetGroupDescription

Description du groupe de sous-réseaux.

Type : chaîne

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBSubnetGroup

Informations détaillées sur un groupe de sous-réseaux.

Type : objet DBSubnetGroup

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBSubnetGroupDoesNotCoverEnoughAZs

Les sous-réseaux du groupe de sous-réseaux doivent couvrir au moins deux zones de disponibilité, sauf s'il n'existe qu'une seule zone de disponibilité.

Code d'état HTTP : 400

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`ne fait pas référence à un groupe de sous-réseaux existant.

Code d'état HTTP : 404

DBSubnetQuotaExceededFault

La demande vous obligerait à dépasser le nombre autorisé de sous-réseaux dans un groupe de sous-réseaux.

Code d'état HTTP : 400

InvalidSubnet

Le sous-réseau demandé n'est pas valide ou plusieurs sous-réseaux ont été demandés mais ils ne se trouvent pas tous dans un cloud privé virtuel (VPC) commun.

Code d'état HTTP : 400

SubnetAlreadyInUse

Le sous-réseau est déjà utilisé dans la zone de disponibilité.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

ModifyEventSubscription

Service : Amazon DocumentDB (with MongoDB compatibility)

Modifie un abonnement existant aux notifications d'événements Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

SubscriptionName

Nom de l'abonnement aux notifications d'événements Amazon DocumentDB.

Type : chaîne

Obligatoire : oui

Enabled

Valeur booléenne ; définie sur pour `true` activer l'abonnement.

Type : booléen

Obligatoire : non

EventCategories. EventCategoryN.

Liste des catégories d'événements `SourceType` auxquels vous souhaitez vous abonner.

Type : tableau de chaînes

Obligatoire : non

SnsTopicArn

Amazon Resource Name (ARN) de la rubrique SNS créé pour la notification d'événements. L'ARN est créé par Amazon SNS lorsque vous créez une rubrique et vous y abonnez.

Type : chaîne

Obligatoire : non

SourceType

Type de source qui génère les événements. Par exemple, si vous souhaitez être informé des événements générés par une instance, définissez ce paramètre sur`db-instance`. Si cette valeur n'est pas spécifiée, tous les événements sont renvoyés.

Valeurs valides: `db-instance`, `db-parameter-group`, `db-security-group`

Type : chaîne

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

EventSubscription

Informations détaillées sur un événement auquel vous vous êtes inscrit.

Type : objet EventSubscription

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

EventSubscriptionQuotaExceeded

Vous avez atteint le nombre maximum d'abonnements aux événements.

Code d’état HTTP : 400

SNSInvalidTopic

Amazon SNS a répondu qu'il y avait un problème avec le sujet spécifié.

Code d’état HTTP : 400

SNSNoAuthorization

Vous n'êtes pas autorisé à publier sur la rubrique SNS Amazon Resource Name (ARN).

Code d’état HTTP : 400

SNSTopicArnNotFound

La rubrique SNS Amazon Resource Name (ARN) n'existe pas.

Code d’état HTTP : 404

SubscriptionCategoryNotFound

La catégorie fournie n'existe pas.

Code d’état HTTP : 404

SubscriptionNotFound

Le nom de l'abonnement n'existe pas.

Code d’état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

ModifyGlobalCluster

Service : Amazon DocumentDB (with MongoDB compatibility)

Modifiez un paramètre pour un cluster global Amazon DocumentDB. Vous pouvez modifier un ou plusieurs paramètres de configuration (par exemple : protection contre les suppressions) ou l'identifiant global du cluster en spécifiant ces paramètres et les nouvelles valeurs dans la demande.

> **Note**
>
> Cette action s'applique uniquement aux clusters Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

GlobalClusterIdentifier

Identifiant du cluster global en cours de modification. Ce paramètre n'est pas sensible à la casse.

Contraintes :

- Doit correspondre à l'identifiant d'un cluster global existant.

Type : chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 255.

Modèle : `[A-Za-z][0-9A-Za-z-:._]*`

Obligatoire : oui

DeletionProtection

Indique si la protection contre les suppressions est activée sur le cluster global. Le cluster global ne peut pas être supprimé lorsque la protection contre la suppression est activée.

Type : booléen

Obligatoire : non

NewGlobalClusterIdentifier

Le nouvel identifiant d'un cluster global lorsque vous modifiez un cluster global. Cette valeur est stockée sous la forme d'une chaîne en minuscules.

- Doit contenir entre 1 et 63 lettres, chiffres ou traits d'union

  Le premier caractère doit être une lettre

  Il ne peut pas se terminer par un trait d'union ou contenir deux traits d'union consécutifs.

Exemple : `my-cluster2`

Type : chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 255.

Modèle : `[A-Za-z][0-9A-Za-z-:._]*`

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

GlobalCluster

Type de données représentant un cluster global Amazon DocumentDB.

Type : objet GlobalCluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

GlobalClusterNotFoundFault

`GlobalClusterIdentifier`Cela ne fait pas référence à un cluster mondial existant.

Code d’état HTTP : 404

InvalidGlobalClusterStateFault

L'opération demandée ne peut pas être effectuée tant que le cluster est dans cet état.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

RebootDBInstance

Service : Amazon DocumentDB (with MongoDB compatibility)

Il se peut que vous deviez redémarrer votre instance, généralement pour des raisons de maintenance. Par exemple, si vous apportez certaines modifications ou si vous modifiez le groupe de paramètres de cluster associé à l'instance, vous devez redémarrer l'instance pour que les modifications prennent effet.

Le redémarrage d'une instance entraîne celui du service du moteur de base de données. Le redémarrage d'une instance entraîne une interruption momentanée, au cours de laquelle le statut de l'instance est défini sur le redémarrage.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBInstanceIdentifier

Identifiant de l’instance. Ce paramètre est stocké sous la forme d'une chaîne en lettres minuscules.

Contraintes :

- Doit correspondre à l'identifiant d'un `DBInstance` existant.

Type : chaîne

Obligatoire : oui

ForceFailover

Lorsque `true` le redémarrage est effectué par le biais d'un basculement multi-AZ.

Contrainte : vous ne pouvez pas spécifier `true` si l'instance n'est pas configurée pour le mode multi-AZ.

Type : booléen

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBInstance

Informations détaillées sur une instance.

Type : objet DBInstance

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBInstanceNotFound

`DBInstanceIdentifier`ne fait pas référence à une instance existante.

Code d'état HTTP : 404

InvalidDBInstanceState

L'instance spécifiée n'est pas disponible.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

RemoveFromGlobalCluster

Service : Amazon DocumentDB (with MongoDB compatibility)

Détache un cluster secondaire Amazon DocumentDB d'un cluster global. Le cluster devient un cluster autonome doté d'une fonction de lecture-écriture au lieu d'être en lecture seule et de recevoir des données d'un serveur principal d'une autre région.

> **Note**
>
> Cette action s'applique uniquement aux clusters Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DbClusterIdentifier

Le nom de ressource Amazon (ARN) identifiant le cluster détaché du cluster global Amazon DocumentDB.

Type : chaîne

Obligatoire : oui

GlobalClusterIdentifier

Identifiant de cluster à détacher du cluster global Amazon DocumentDB.

Type : chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 255.

Modèle : `[A-Za-z][0-9A-Za-z-:._]*`

Obligatoire : oui

Éléments de réponse

L'élément suivant est renvoyé par le service.

GlobalCluster

Type de données représentant un cluster global Amazon DocumentDB.

Type : objet GlobalCluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterNotFoundFault

`DBClusterIdentifier`ne fait pas référence à un cluster existant.

Code d'état HTTP : 404

GlobalClusterNotFoundFault

`GlobalClusterIdentifier`Cela ne fait pas référence à un cluster mondial existant.

Code d'état HTTP : 404

InvalidGlobalClusterStateFault

L'opération demandée ne peut pas être effectuée tant que le cluster est dans cet état.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3

- AWS SDK pour Python
- AWS SDK pour Ruby V3

RemoveSourceIdentifierFromSubscription

Service : Amazon DocumentDB (with MongoDB compatibility)

Supprime un identifiant de source d'un abonnement existant aux notifications d'événements Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

SourceIdentifier

L'identifiant source à supprimer de l'abonnement, tel que l'identifiant d'instance d'une instance ou le nom d'un groupe de sécurité.

Type : chaîne

Obligatoire : oui

SubscriptionName

Nom de l'abonnement aux notifications d'événements Amazon DocumentDB dont vous souhaitez supprimer un identifiant de source.

Type : chaîne

Obligatoire : oui

Éléments de réponse

L'élément suivant est renvoyé par le service.

EventSubscription

Informations détaillées sur un événement auquel vous vous êtes inscrit.

Type : objet EventSubscription

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

SourceNotFound

La source demandée n'a pas pu être trouvée.

Code d’état HTTP : 404

SubscriptionNotFound

Le nom de l'abonnement n'existe pas.

Code d’état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

RemoveTagsFromResource

Service : Amazon DocumentDB (with MongoDB compatibility)

Supprime les balises de métadonnées d'une ressource Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

ResourceName

La ressource Amazon DocumentDB dont les balises sont supprimées. Cette valeur est un Amazon Resource Name (ARN).

Type : chaîne

Obligatoire : oui

TagKeys.membre.n

Clé de balise (nom) de la balise à supprimer.

Type : tableau de chaînes

Obligatoire : oui

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterNotFoundFault

`DBClusterIdentifier`ne fait pas référence à un cluster existant.

Code d'état HTTP : 404

DBInstanceNotFound

`DBInstanceIdentifier`ne fait pas référence à une instance existante.

Code d'état HTTP : 404

DBSnapshotNotFound

`DBSnapshotIdentifier`ne fait pas référence à un instantané existant.

Code d'état HTTP : 404

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

ResetDBClusterParameterGroup

Service : Amazon DocumentDB (with MongoDB compatibility)

Modifie les paramètres d'un groupe de paramètres de cluster à la valeur par défaut. Pour réinitialiser des paramètres spécifiques, soumettez une liste des éléments suivants : `ParameterName` et`ApplyMethod`. Pour réinitialiser l'ensemble du groupe de paramètres du cluster, spécifiez les `ResetAllParameters` paramètres `DBClusterParameterGroupName` et.

Lorsque vous réinitialisez l'ensemble du groupe, les paramètres dynamiques sont immédiatement mis à jour et les paramètres statiques sont définis pour `pending-reboot` prendre effet au prochain redémarrage de l'instance de base de données.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterParameterGroupName

Le nom du groupe de paramètres du cluster à réinitialiser.

Type : chaîne

Obligatoire : oui

Paramètres.Paramètre.N

Liste des noms de paramètres du groupe de paramètres du cluster à rétablir aux valeurs par défaut. Vous ne pouvez pas utiliser ce paramètre si le paramètre `ResetAllParameters` est défini sur `true`.

Type : tableau d’objets Parameter

Obligatoire : non

ResetAllParameters

Valeur définie de manière `true` à rétablir les valeurs par défaut de tous les paramètres du groupe de paramètres du cluster, et dans le `false` cas contraire. Vous ne pouvez pas utiliser ce paramètre si une liste des noms des paramètres spécifiés existe pour le paramètre `Parameters`.

Type : booléen

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBClusterParameterGroupName

Nom d'un groupe de paramètres de cluster.

Contraintes :

- Doit contenir de 1 à 255 lettres ou chiffres.
- Le premier caractère doit être une lettre.
- Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

> **Note**
>
> Cette valeur est stockée sous la forme d'une chaîne en minuscules.

Type : chaîne

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBParameterGroupNotFound

`DBParameterGroupName`ne fait pas référence à un groupe de paramètres existant.

Code d'état HTTP : 404

InvalidDBParameterGroupState

Le groupe de paramètres est en cours d'utilisation ou son état n'est pas valide. Si vous essayez de supprimer le groupe de paramètres, vous ne pouvez pas le supprimer lorsque le groupe de paramètres est dans cet état.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

RestoreDBClusterFromSnapshot

Service : Amazon DocumentDB (with MongoDB compatibility)

Crée un nouveau cluster à partir d'un instantané ou d'un instantané de cluster.

Si un instantané est spécifié, le cluster cible est créé à partir de l'instantané de base de données source avec une configuration et un groupe de sécurité par défaut.

Si un instantané de cluster est spécifié, le cluster cible est créé à partir du point de restauration du cluster source avec la même configuration que le cluster de base de données source d'origine, sauf que le nouveau cluster est créé avec le groupe de sécurité par défaut.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterIdentifier

Nom du cluster à créer à partir du cliché ou de l'instantané du cluster. Ce paramètre n'est pas sensible à la casse.

Contraintes :

- Doit contenir entre 1 et 63 lettres, chiffres ou traits d'union.
- Le premier caractère doit être une lettre.
- Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

Exemple : `my-snapshot-id`

Type : chaîne

Obligatoire : oui

Engine

Le moteur de base de données à utiliser pour le nouveau cluster.

Par défaut : identique à la source.

Contrainte : Doit être compatible avec le moteur de la source.

Type : chaîne

Obligatoire : oui

SnapshotIdentifier

Identifiant de l'instantané ou de l'instantané de cluster à partir duquel effectuer la restauration.

Vous pouvez utiliser le nom ou l'Amazon Resource Name (ARN) pour spécifier un instantané de cluster. Cependant, vous pouvez uniquement utiliser l'ARN pour spécifier un instantané.

Contraintes :

- Doit correspondre à l'identifiant d'un instantané existant.

Type : chaîne

Obligatoire : oui

AvailabilityZones. AvailabilityZoneN.

Fournit la liste des zones de disponibilité Amazon EC2 dans lesquelles les instances du cluster de base de données restauré peuvent être créées.

Type : tableau de chaînes

Obligatoire : non

DBClusterParameterGroupName

Nom du groupe de paramètres de cluster DB à associer à ce cluster DB.

Type : chaîne. Nécessaire : Non

Si cet argument est omis, le groupe de paramètres du cluster de base de données par défaut est utilisé. S'il est fourni, il doit correspondre au nom d'un groupe de paramètres de cluster de base de données par défaut existant. La chaîne doit être composée de 1 à 255 lettres, chiffres ou tirets. Son premier caractère doit être une lettre et il ne peut pas se terminer par un tiret ni contenir deux tirets consécutifs.

Type : chaîne

Obligatoire : non

DBSubnetGroupName

Nom du groupe de sous-réseaux à utiliser pour le nouveau cluster.

Contraintes : si indiqué, il doit correspondre au nom d'un existant`DBSubnetGroup`.

Exemple : `mySubnetgroup`

Type : chaîne

Obligatoire : non

DeletionProtection

Spécifie si ce cluster peut être supprimé. Si cette option `DeletionProtection` est activée, le cluster ne peut pas être supprimé sauf s'il `DeletionProtection` est modifié et désactivé. `DeletionProtection`protège les clusters contre la suppression accidentelle.

Type : booléen

Obligatoire : non

EnableCloudwatchLogsExports.membre.n

Liste des types de journaux qui doivent être activés pour être exportés vers Amazon CloudWatch Logs.

Type : tableau de chaînes

Obligatoire : non

EngineVersion

Version du moteur de base de données à utiliser pour le nouveau cluster.

Type : chaîne

Obligatoire : non

KmsKeyId

Identifiant de AWS KMS clé à utiliser lors de la restauration d'un cluster chiffré à partir d'un instantané de base de données ou d'un instantané de cluster.

L'identifiant de AWS KMS clé est le Amazon Resource Name (ARN) de la clé de AWS KMS chiffrement. Si vous restaurez un cluster Compte AWS possédant la clé de AWS KMS chiffrement utilisée pour chiffrer le nouveau cluster, vous pouvez utiliser l'alias de AWS KMS clé au lieu de l'ARN pour la clé de AWS KMS chiffrement.

Si vous ne spécifiez pas de valeur pour le paramètre `KmsKeyId` :

- Si le cliché ou le cliché de cluster `SnapshotIdentifier` intégré est chiffré, le cluster restauré est chiffré à l'aide de la AWS KMS clé qui a été utilisée pour chiffrer le cliché ou le cliché de cluster.
- Si le snapshot ou le snapshot du cluster n'`SnapshotIdentifier`est pas chiffré, le cluster de base de données restauré n'est pas chiffré.

Type : chaîne

Obligatoire : non

Port

Numéro de port sur lequel le nouveau cluster accepte les connexions.

Contraintes : Doit être une valeur comprise entre `1150` et`65535`.

Par défaut : le même port que le cluster d'origine.

Type : entier

Obligatoire : non

StorageType

Type de stockage à associer au cluster de base de données.

Pour plus d'informations sur les types de stockage pour les clusters Amazon DocumentDB, consultez la section Configurations de stockage des clusters dans le manuel Amazon DocumentDB Developer Guide.

Valeurs valides pour le type de stockage - `standard | iopt1`

La valeur par défaut est `standard`

Type : chaîne

Obligatoire : non

Étiquettes.Tag.N

Les balises à attribuer au cluster restauré.

Type : tableau d’objets Tag

Obligatoire : non

VpcSecurityGroupIds. VpcSecurityGroupIdN.

Liste des groupes de sécurité du cloud privé virtuel (VPC) auxquels le nouveau cluster appartiendra.

Type : tableau de chaînes

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBCluster

Informations détaillées sur un cluster.

Type : objet DBCluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterAlreadyExistsFault

Vous avez déjà un cluster avec l'identifiant indiqué.

Code d’état HTTP : 400

DBClusterQuotaExceededFault

Le cluster ne peut pas être créé car vous avez atteint le quota maximum autorisé de clusters.

Code d’état HTTP : 403

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`ne fait pas référence à un instantané de cluster existant.

Code d’état HTTP : 404

DBSnapshotNotFound

DBSnapshotIdentifierne fait pas référence à un instantané existant.

Code d'état HTTP : 404

DBSubnetGroupNotFoundFault

DBSubnetGroupNamene fait pas référence à un groupe de sous-réseaux existant.

Code d'état HTTP : 404

DBSubnetGroupNotFoundFault

DBSubnetGroupNamene fait pas référence à un groupe de sous-réseaux existant.

Code d'état HTTP : 404

InsufficientDBClusterCapacityFault

Le cluster ne dispose pas d'une capacité suffisante pour l'opération en cours.

Code d'état HTTP : 403

InsufficientStorageClusterCapacity

L'espace de stockage disponible est insuffisant pour l'action en cours. Vous pouvez peut-être résoudre cette erreur en mettant à jour votre groupe de sous-réseaux afin qu'il utilise différentes zones de disponibilité disposant d'un espace de stockage plus important.

Code d'état HTTP : 400

InvalidDBClusterSnapshotStateFault

La valeur fournie n'est pas un état de capture d'écran de cluster valide.

Code d'état HTTP : 400

InvalidDBSnapshotState

L'état de l'instantané ne permet pas de le supprimer.

Code d'état HTTP : 400

InvalidRestoreFault

Vous ne pouvez pas effectuer de restauration à partir d'une sauvegarde de cloud privé virtuel (VPC) vers une instance de base de données non VPC.

Code d’état HTTP : 400

InvalidSubnet

Le sous-réseau demandé n'est pas valide ou plusieurs sous-réseaux ont été demandés mais ils ne se trouvent pas tous dans un cloud privé virtuel (VPC) commun.

Code d’état HTTP : 400

InvalidVPCNetworkStateFault

Le groupe de sous-réseaux ne couvre pas toutes les zones de disponibilité après sa création en raison des modifications apportées.

Code d’état HTTP : 400

KMSKeyNotAccessibleFault

Une erreur s'est produite lors de l'accès à une AWS KMS clé.

Code d’état HTTP : 400

StorageQuotaExceeded

La demande vous obligerait à dépasser la quantité de stockage autorisée disponible sur toutes les instances.

Code d’état HTTP : 400

StorageQuotaExceeded

La demande vous obligerait à dépasser la quantité de stockage autorisée disponible sur toutes les instances.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++

- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

RestoreDBClusterToPointInTime

Service : Amazon DocumentDB (with MongoDB compatibility)

Restaure un cluster à un moment arbitraire. Les utilisateurs peuvent restaurer à tout moment avant `LatestRestorableTime` pendant `BackupRetentionPeriod` jours. Le cluster cible est créé à partir du cluster source avec la même configuration que le cluster d'origine, sauf que le nouveau cluster est créé avec le groupe de sécurité par défaut.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterIdentifier

Il faut créer le nom du nouveau cluster.

Contraintes :

- Doit contenir entre 1 et 63 lettres, chiffres ou traits d'union.
- Le premier caractère doit être une lettre.
- Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

Type : chaîne

Obligatoire : oui

SourceDBClusterIdentifier

L'identifiant du cluster source à partir duquel effectuer la restauration.

Contraintes :

- Doit correspondre à l'identifiant d'un `DBCluster` existant.

Type : chaîne

Obligatoire : oui

DBSubnetGroupName

Le nom du groupe de sous-réseaux à utiliser pour le nouveau cluster.

Contraintes : si indiqué, il doit correspondre au nom d'un existant`DBSubnetGroup`.

Exemple : `mySubnetgroup`

Type : chaîne

Obligatoire : non

DeletionProtection

Spécifie si ce cluster peut être supprimé. Si cette option `DeletionProtection` est activée, le cluster ne peut pas être supprimé sauf s'il `DeletionProtection` est modifié et désactivé. `DeletionProtection`protège les clusters contre la suppression accidentelle.

Type : booléen

Obligatoire : non

EnableCloudwatchLogsExports.membre.n

Liste des types de journaux qui doivent être activés pour être exportés vers Amazon CloudWatch Logs.

Type : tableau de chaînes

Obligatoire : non

KmsKeyId

Identifiant de AWS KMS clé à utiliser lors de la restauration d'un cluster chiffré à partir d'un cluster chiffré.

L'identifiant de AWS KMS clé est le Amazon Resource Name (ARN) de la clé de AWS KMS chiffrement. Si vous restaurez un cluster Compte AWS possédant la clé de AWS KMS chiffrement utilisée pour chiffrer le nouveau cluster, vous pouvez utiliser l'alias de AWS KMS clé au lieu de l'ARN pour la clé de AWS KMS chiffrement.

Vous pouvez effectuer une restauration sur un nouveau cluster et chiffrer le nouveau cluster avec une AWS KMS clé différente de celle utilisée pour chiffrer le cluster source. AWS KMS Le nouveau cluster de base de données est chiffré avec la AWS KMS clé identifiée par le `KmsKeyId` paramètre.

Si vous ne spécifiez pas de valeur pour le paramètre `KmsKeyId` :

- Si le cluster est chiffré, le cluster restauré est chiffré à l'aide de la AWS KMS clé utilisée pour chiffrer le cluster source.

- Si le cluster n'est pas chiffré, le cluster restauré n'est pas chiffré.

S'il s'`DBClusterIdentifier`agit d'un cluster non chiffré, la demande de restauration est rejetée.

Type : chaîne

Obligatoire : non

Port

Numéro de port sur lequel le nouveau cluster accepte les connexions.

Contraintes : Doit être une valeur comprise entre `1150` et`65535`.

Par défaut : port par défaut du moteur.

Type : entier

Obligatoire : non

RestoreToTime

La date et l'heure auxquelles restaurer le cluster.

Valeurs valides : une heure au format UTC (temps universel)

Contraintes :

- Doit être antérieure à la dernière date de restauration de l'instance.
- Cela doit être indiqué si le paramètre `UseLatestRestorableTime` n'est pas fourni.
- Cela ne peut pas être spécifié lorsque le paramètre `UseLatestRestorableTime` est `true`.
- Cela ne peut pas être spécifié lorsque le paramètre `RestoreType` est `copy-on-write`.

Exemple : `2015-03-07T23:45:00Z`

Type : Timestamp

Obligatoire : non

RestoreType

Type de restauration à exécuter. Vous pouvez spécifier l’une des valeurs suivantes :

- `full-copy` - Le nouveau cluster de base de données est restauré sous la forme d'une copie intégrale du cluster de base de données source.

- `copy-on-write` - Le nouveau cluster de base de données est restauré sous la forme d'un clone du cluster de base de données source.

Contraintes : vous ne pouvez pas spécifier `copy-on-write` si la version du moteur du cluster de base de données source est antérieure à la version 1.11.

Si vous ne spécifiez pas de valeur pour `RestoreType`, le nouveau cluster de base de données est restauré sous la forme d'une copie intégrale du cluster de base de données source.

Type : chaîne

Obligatoire : non

StorageType

Type de stockage à associer au cluster de base de données.

Pour plus d'informations sur les types de stockage pour les clusters Amazon DocumentDB, consultez la section Configurations de stockage des clusters dans le manuel Amazon DocumentDB Developer Guide.

Valeurs valides pour le type de stockage - `standard | iopt1`

La valeur par défaut est `standard`

Type : chaîne

Obligatoire : non

Étiquettes.Tag.N

Les balises à attribuer au cluster restauré.

Type : tableau d’objets Tag

Obligatoire : non

UseLatestRestorableTime

Une valeur définie sur `true` pour restaurer le cluster à la dernière heure de sauvegarde restaurable, et sur `false` dans le cas contraire.

Par défaut : `false`

Contraintes : ne peut pas être spécifiée si le paramètre `RestoreToTime` est fourni.

Type : booléen

Obligatoire : non

VpcSecurityGroupIds. VpcSecurityGroupIdN.

Liste des groupes de sécurité VPC auxquels appartient le nouveau cluster.

Type : tableau de chaînes

Obligatoire : non

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBCluster

Informations détaillées sur un cluster.

Type : objet DBCluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterAlreadyExistsFault

Vous avez déjà un cluster avec l'identifiant indiqué.

Code d'état HTTP : 400

DBClusterNotFoundFault

`DBClusterIdentifier`ne fait pas référence à un cluster existant.

Code d'état HTTP : 404

DBClusterQuotaExceededFault

Le cluster ne peut pas être créé car vous avez atteint le quota maximum autorisé de clusters.

Code d'état HTTP : 403

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`ne fait pas référence à un instantané de cluster existant.

Code d'état HTTP : 404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`ne fait pas référence à un groupe de sous-réseaux existant.

Code d'état HTTP : 404

InsufficientDBClusterCapacityFault

Le cluster ne dispose pas d'une capacité suffisante pour l'opération en cours.

Code d'état HTTP : 403

InsufficientStorageClusterCapacity

L'espace de stockage disponible est insuffisant pour l'action en cours. Vous pouvez peut-être résoudre cette erreur en mettant à jour votre groupe de sous-réseaux afin qu'il utilise différentes zones de disponibilité disposant d'un espace de stockage plus important.

Code d'état HTTP : 400

InvalidDBClusterSnapshotStateFault

La valeur fournie n'est pas un état de capture d'écran de cluster valide.

Code d'état HTTP : 400

InvalidDBClusterStateFault

Le cluster n'est pas dans un état valide.

Code d'état HTTP : 400

InvalidDBSnapshotState

L'état de l'instantané ne permet pas de le supprimer.

Code d'état HTTP : 400

InvalidRestoreFault

Vous ne pouvez pas effectuer de restauration à partir d'une sauvegarde de cloud privé virtuel (VPC) vers une instance de base de données non VPC.

Code d'état HTTP : 400

InvalidSubnet

Le sous-réseau demandé n'est pas valide ou plusieurs sous-réseaux ont été demandés mais ils ne se trouvent pas tous dans un cloud privé virtuel (VPC) commun.

Code d'état HTTP : 400

InvalidVPCNetworkStateFault

Le groupe de sous-réseaux ne couvre pas toutes les zones de disponibilité après sa création en raison des modifications apportées.

Code d'état HTTP : 400

KMSKeyNotAccessibleFault

Une erreur s'est produite lors de l'accès à une AWS KMS clé.

Code d'état HTTP : 400

StorageQuotaExceeded

La demande vous obligerait à dépasser la quantité de stockage autorisée disponible sur toutes les instances.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python

- AWS SDK pour Ruby V3

StartDBCluster

Service : Amazon DocumentDB (with MongoDB compatibility)

Redémarre le cluster arrêté spécifié par`DBClusterIdentifier`. Pour plus d'informations, consultez Arrêter et démarrer un cluster Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterIdentifier

Identifiant du cluster à redémarrer. Exemple : `docdb-2019-05-28-15-24-52`

Type : chaîne

Obligatoire : oui

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBCluster

Informations détaillées sur un cluster.

Type : objet DBCluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterNotFoundFault

`DBClusterIdentifier`ne fait pas référence à un cluster existant.

Code d’état HTTP : 404

InvalidDBClusterStateFault

Le cluster n'est pas dans un état valide.

Code d’état HTTP : 400

InvalidDBInstanceState

L'instance spécifiée n'est pas dans l'état disponible.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

StopDBCluster

Service : Amazon DocumentDB (with MongoDB compatibility)

Arrête le cluster en cours d'exécution spécifié par`DBClusterIdentifier`. Le cluster doit être dans l'état disponible. Pour plus d'informations, consultez Arrêter et démarrer un cluster Amazon DocumentDB.

Paramètres de demande

Pour plus d'informations sur les paramètres courants pour toutes les actions, consultez Paramètres courants.

DBClusterIdentifier

Identifiant du cluster à arrêter. Exemple : `docdb-2019-05-28-15-24-52`

Type : chaîne

Obligatoire : oui

Éléments de réponse

L'élément suivant est renvoyé par le service.

DBCluster

Informations détaillées sur un cluster.

Type : objet DBCluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

DBClusterNotFoundFault

`DBClusterIdentifier`ne fait pas référence à un cluster existant.

Code d'état HTTP : 404

InvalidDBClusterStateFault

Le cluster n'est pas dans un état valide.

Code d’état HTTP : 400

InvalidDBInstanceState

L'instance spécifiée n'est pas dans l'état disponible.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

Clusters Amazon DocumentDB Elastic

Les actions suivantes sont prises en charge par Amazon DocumentDB Elastic Clusters :

- CopyClusterSnapshot
- CreateCluster
- CreateClusterSnapshot
- DeleteCluster
- DeleteClusterSnapshot
- GetCluster

- GetClusterSnapshot
- ListClusters
- ListClusterSnapshots
- ListTagsForResource
- RestoreClusterFromSnapshot
- StartCluster
- StopCluster
- TagResource
- UntagResource
- UpdateCluster

CopyClusterSnapshot

Service : Amazon DocumentDB Elastic Clusters

Copie un instantané d'un cluster élastique.

Syntaxe de la demande

```
POST /cluster-snapshot/snapshotArn/copy HTTP/1.1
Content-type: application/json

{
   "copyTags": boolean,
   "kmsKeyId": "string",
   "tags": {
      "string" : "string"
   },
   "targetSnapshotName": "string"
}
```

Paramètres de demande URI

La demande utilise les paramètres URI suivants.

snapshotArn

L'identifiant Amazon Resource Name (ARN) de l'instantané du cluster élastique.

Obligatoire : oui

Corps de la demande

Cette demande accepte les données suivantes au format JSON.

targetSnapshotName

Identifiant du nouvel instantané de cluster élastique à créer à partir de l'instantané de cluster source. Ce paramètre n'est pas sensible à la casse.

Contraintes :

- Doit contenir entre 1 et 63 lettres, chiffres ou traits d'union.
- Le premier caractère doit être une lettre.

- Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

Exemple : `elastic-cluster-snapshot-5`

Type : chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Obligatoire : oui

copyTags

Définissez sur `true` pour copier toutes les balises de l'instantané du cluster source vers l'instantané du cluster Elastic cible. L'argument par défaut est `false`.

Type : booléen

Obligatoire : non

kmsKeyId

ID de clé AWS KMS pour un instantané de cluster élastique chiffré. L'ID de clé AWS KMS est le nom de ressource Amazon (ARN), l'identifiant de clé AWS KMS ou l'alias de clé AWS KMS pour la clé de chiffrement AWS KMS.

Si vous copiez un instantané chiffré du cluster Elastic depuis votre AWS compte, vous pouvez spécifier une valeur `KmsKeyId` pour chiffrer la copie avec une nouvelle clé de chiffrement AWS S KMS. Si vous ne spécifiez aucune valeur pour`KmsKeyId`, la copie de l'instantané du cluster élastique est chiffrée avec la même clé `AWS` KMS que l'instantané du cluster élastique source.

Pour copier un instantané de cluster élastique chiffré vers une autre AWS région, définissez `KmsKeyId` l'ID de clé AWS KMS que vous souhaitez utiliser pour chiffrer la copie de l'instantané de cluster élastique dans la région de destination. AWS Les clés de chiffrement KMS sont spécifiques à la AWS région dans laquelle elles sont créées, et vous ne pouvez pas utiliser les clés de chiffrement d'une AWS région dans une autre AWS .

Si vous copiez un instantané d'un cluster élastique non chiffré et que vous spécifiez une valeur pour le `KmsKeyId` paramètre, une erreur est renvoyée.

Type : chaîne

Obligatoire : non

tags

Les balises à attribuer à l'instantané du cluster élastique.

Type : mappage chaîne/chaîne

Contraintes de longueur de clé : longueur minimale de 1. Longueur maximale de 128.

Modèle de clé :`^(?!aws:)[a-zA-Z+-=._:/]+$`

Contraintes de longueur de valeur : longueur minimale de 0. Longueur maximale de 256.

Obligatoire : non

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "snapshot": {
      "adminUserName": "string",
      "clusterArn": "string",
      "clusterCreationTime": "string",
      "kmsKeyId": "string",
      "snapshotArn": "string",
      "snapshotCreationTime": "string",
      "snapshotName": "string",
      "snapshotType": "string",
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Eléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

snapshot

Renvoie des informations sur un instantané de cluster élastique spécifique.

Type : objet ClusterSnapshot

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

AccessDeniedException

Exception qui se produit lorsque les autorisations ne sont pas suffisantes pour effectuer une action.

Code d’état HTTP : 403

ConflictException

Il y a eu un conflit d'accès.

Code d’état HTTP : 409

InternalServerException

Une erreur interne du serveur s'est produite.

Code d’état HTTP : 500

ResourceNotFoundException

La ressource spécifiée n'a pas pu être localisée.

Code d’état HTTP : 404

ServiceQuotaExceededException

Le quota de service pour l'action a été dépassé.

Code d'état HTTP : 402

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d’état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

CreateCluster

Service : Amazon DocumentDB Elastic Clusters

Crée un nouveau cluster élastique Amazon DocumentDB et renvoie sa structure de cluster.

Syntaxe de la demande

```
POST /cluster HTTP/1.1
Content-type: application/json

{
   "adminUserName": "string",
   "adminUserPassword": "string",
   "authType": "string",
   "backupRetentionPeriod": number,
   "clientToken": "string",
   "clusterName": "string",
   "kmsKeyId": "string",
   "preferredBackupWindow": "string",
   "preferredMaintenanceWindow": "string",
   "shardCapacity": number,
   "shardCount": number,
   "shardInstanceCount": number,
   "subnetIds": [ "string" ],
   "tags": {
      "string" : "string"
   },
   "vpcSecurityGroupIds": [ "string" ]
}
```

Paramètres de demande URI

La demande n’utilise pas de paramètres URI.

Corps de la demande

Cette demande accepte les données suivantes au format JSON.

adminUserName

Nom de l'administrateur des clusters élastiques Amazon DocumentDB.

Contraintes :

- Doit comporter entre 1 et 63 lettres ou chiffres.
- Le premier caractère doit être une lettre.
- Ne peut pas être un mot réservé.

Type : chaîne

Obligatoire : oui

adminUserPassword

Le mot de passe de l'administrateur des clusters élastiques Amazon DocumentDB. Le mot de passe peut contenir n'importe quel caractère ASCII imprimable.

Contraintes :

- Doit contenir de 8 à 100 caractères.
- Ne peut pas contenir de barre oblique (/), de guillemet double («) ou le symbole « at » (@).

Type : chaîne

Obligatoire : oui

authType

Type d'authentification utilisé pour déterminer où récupérer le mot de passe utilisé pour accéder au cluster élastique. Les types valides sont `PLAIN_TEXT` ou`SECRET_ARN`.

Type : chaîne

Valeurs valides : `PLAIN_TEXT | SECRET_ARN`

Obligatoire : oui

clusterName

Nom du nouveau cluster élastique. Ce paramètre est stocké sous la forme d'une chaîne en lettres minuscules.

Contraintes :

- Doit contenir entre 1 et 63 lettres, chiffres ou traits d'union.
- Le premier caractère doit être une lettre.
- Ne peut pas se terminer par un trait d'union ni contenir deux traits d'union consécutifs.

Exemple : `my-cluster`

Type : chaîne

Obligatoire : oui

shardCapacity

Le nombre de vCPU assignés à chaque partition de cluster élastique. Le maximum est de 64. Les valeurs autorisées sont 2, 4, 8, 16, 32, 64.

Type : entier

Obligatoire : oui

shardCount

Le nombre de partitions attribuées au cluster élastique. Le maximum est de 32.

Type : entier

Obligatoire : oui

backupRetentionPeriod

Nombre de jours pendant lesquels les instantanés automatiques sont conservés.

Type : entier

Obligatoire : non

clientToken

Le jeton client pour le cluster élastique.

Type : chaîne

Obligatoire : non

kmsKeyId

Identifiant de clé KMS à utiliser pour chiffrer le nouveau cluster élastique.

L'identifiant de clé KMS est l'Amazon Resource Name (ARN) de la clé de chiffrement KMS. Si vous créez un cluster en utilisant le même compte Amazon qui possède cette clé de chiffrement KMS, vous pouvez utiliser l'alias de clé KMS au lieu de l'ARN comme clé de chiffrement KMS.

Si aucune clé de chiffrement n'est spécifiée, Amazon DocumentDB utilise la clé de chiffrement par défaut créée par KMS pour votre compte. Votre compte possède une clé de chiffrement par défaut différente pour chaque région Amazon.

Type : chaîne

Obligatoire : non

preferredBackupWindow

La plage horaire quotidienne pendant laquelle les sauvegardes automatisées sont créées si les sauvegardes automatisées sont activées, comme déterminé par le`backupRetentionPeriod`.

Type : chaîne

Obligatoire : non

preferredMaintenanceWindow

Intervalle de temps hebdomadaire, au format Universal Coordinated Time (UTC), pendant lequel a lieu la maintenance du système.

Format : `ddd:hh24:mi-ddd:hh24:mi`

Par défaut : une fenêtre de 30 minutes sélectionnée au hasard dans un intervalle de 8 heures pour chacune d'elles Région AWS, survenant un jour aléatoire de la semaine.

Jours valides : lundi, mardi, mercredi, jeudi, vendredi, samedi, dimanche

Contraintes : fenêtre minimale de 30 minutes.

Type : chaîne

Obligatoire : non

shardInstanceCount

Le nombre d'instances de répliques s'appliquant à toutes les partitions du cluster élastique. Une `shardInstanceCount` valeur de 1 signifie qu'il existe une instance d'écriture et que toutes les instances supplémentaires sont des répliques qui peuvent être utilisées pour les lectures et pour améliorer la disponibilité.

Type : entier

Obligatoire : non

subnetIds

Les identifiants de sous-réseau Amazon EC2 pour le nouveau cluster élastique.

Type : tableau de chaînes

Obligatoire : non

tags

Les balises à attribuer au nouveau cluster élastique.

Type : mappage chaîne/chaîne

Contraintes de longueur de clé : longueur minimale de 1. Longueur maximale de 128.

Modèle de clé :`^(?!aws:)[a-zA-Z+-=._:/]+$`

Contraintes de longueur de valeur : longueur minimale de 0. Longueur maximale de 256.

Obligatoire : non

vpcSecurityGroupIds

Liste des groupes de sécurité VPC EC2 à associer au nouveau cluster élastique.

Type : tableau de chaînes

Obligatoire : non

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
```

```
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

cluster

Le nouveau cluster élastique qui a été créé.

Type : objet Cluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

AccessDeniedException

Exception qui se produit lorsque les autorisations ne sont pas suffisantes pour effectuer une action.

Code d'état HTTP : 403

ConflictException

Il y a eu un conflit d'accès.

Code d’état HTTP : 409

InternalServerException

Une erreur interne s'est produite au niveau du serveur.

Code d’état HTTP : 500

ServiceQuotaExceededException

Le quota de service pour l'action a été dépassé.

Code d'état HTTP : 402

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d’état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3

- AWS SDK pour Python
- AWS SDK pour Ruby V3

CreateClusterSnapshot

Service : Amazon DocumentDB Elastic Clusters

Crée un instantané d'un cluster élastique.

Syntaxe de la demande

```
POST /cluster-snapshot HTTP/1.1
Content-type: application/json

{
   "clusterArn": "string",
   "snapshotName": "string",
   "tags": {
      "string" : "string"
   }
}
```

Paramètres de demande URI

La demande n'utilise pas de paramètres URI.

Corps de la demande

Cette demande accepte les données suivantes au format JSON.

clusterArn

Identifiant ARN du cluster élastique dont vous souhaitez créer un instantané.

Type : chaîne

Obligatoire : oui

snapshotName

Nom du nouvel instantané du cluster élastique.

Type : chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximum de 63.

Obligatoire : oui

tags

Les balises à attribuer au nouvel instantané du cluster élastique.

Type : mappage chaîne/chaîne

Contraintes de longueur de clé : longueur minimale de 1. Longueur maximale de 128.

Modèle de clé :`^(?!aws:)[a-zA-Z+-=._:/]+$`

Contraintes de longueur de valeur : longueur minimale de 0. Longueur maximale de 256.

Obligatoire : non

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "snapshot": {
      "adminUserName": "string",
      "clusterArn": "string",
      "clusterCreationTime": "string",
      "kmsKeyId": "string",
      "snapshotArn": "string",
      "snapshotCreationTime": "string",
      "snapshotName": "string",
      "snapshotType": "string",
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

snapshot

Renvoie des informations sur le nouvel instantané du cluster élastique.

Type : objet ClusterSnapshot

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

AccessDeniedException

Exception qui se produit lorsque les autorisations ne sont pas suffisantes pour effectuer une action.

Code d’état HTTP : 403

ConflictException

Il y a eu un conflit d'accès.

Code d’état HTTP : 409

InternalServerException

Une erreur interne s'est produite au niveau du serveur.

Code d’état HTTP : 500

ResourceNotFoundException

La ressource spécifiée n'a pas pu être localisée.

Code d’état HTTP : 404

ServiceQuotaExceededException

Le quota de service pour l'action a été dépassé.

Code d'état HTTP : 402

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d’état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DeleteCluster

Service : Amazon DocumentDB Elastic Clusters

Supprimez un cluster élastique.

Syntaxe de la demande

```
DELETE /cluster/clusterArn HTTP/1.1
```

Paramètres de demande URI

La demande utilise les paramètres URI suivants.

clusterArn

Identifiant ARN du cluster élastique à supprimer.

Obligatoire : oui

Corps de la demande

La demande n’a pas de corps de requête.

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
```

```
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

cluster

Renvoie des informations sur le cluster élastique récemment supprimé.

Type : objet Cluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

AccessDeniedException

Exception qui se produit lorsque les autorisations ne sont pas suffisantes pour effectuer une action.

Code d'état HTTP : 403

ConflictException

Il y a eu un conflit d'accès.

Code d'état HTTP : 409

InternalServerException

Une erreur interne du serveur s'est produite.

Code d’état HTTP : 500

ResourceNotFoundException

La ressource spécifiée n'a pas pu être localisée.

Code d’état HTTP : 404

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d’état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

DeleteClusterSnapshot

Service : Amazon DocumentDB Elastic Clusters

Supprimez un instantané d'Elastic Cluster.

Syntaxe de la demande

```
DELETE /cluster-snapshot/snapshotArn HTTP/1.1
```

Paramètres de demande URI

La demande utilise les paramètres URI suivants.

snapshotArn

Identifiant ARN de l'instantané du cluster élastique à supprimer.

Obligatoire : oui

Corps de la demande

La demande n’a pas de corps de requête.

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "snapshot": {
      "adminUserName": "string",
      "clusterArn": "string",
      "clusterCreationTime": "string",
      "kmsKeyId": "string",
      "snapshotArn": "string",
      "snapshotCreationTime": "string",
      "snapshotName": "string",
      "snapshotType": "string",
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
```

```
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

snapshot

Renvoie des informations sur le cliché du cluster élastique récemment supprimé.

Type : objet ClusterSnapshot

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

AccessDeniedException

Exception qui se produit lorsque les autorisations ne sont pas suffisantes pour effectuer une action.

Code d'état HTTP : 403

ConflictException

Il y a eu un conflit d'accès.

Code d'état HTTP : 409

InternalServerException

Une erreur interne du serveur s'est produite.

Code d'état HTTP : 500

ResourceNotFoundException

La ressource spécifiée n'a pas pu être localisée.

Code d'état HTTP : 404

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d'état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

GetCluster

Service : Amazon DocumentDB Elastic Clusters

Renvoie des informations sur un cluster élastique spécifique.

Syntaxe de la demande

```
GET /cluster/clusterArn HTTP/1.1
```

Paramètres de demande URI

La demande utilise les paramètres URI suivants.

clusterArn

L'identifiant ARN du cluster élastique.

Obligatoire : oui

Corps de la demande

La demande n’a pas de corps de requête.

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
```

```
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

cluster

Renvoie des informations sur un cluster élastique spécifique.

Type : objet Cluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

AccessDeniedException

Exception qui se produit lorsque les autorisations ne sont pas suffisantes pour effectuer une action.

Code d'état HTTP : 403

InternalServerException

Une erreur interne s'est produite au niveau du serveur.

Code d'état HTTP : 500

ResourceNotFoundException

La ressource spécifiée n'a pas pu être localisée.

Code d’état HTTP : 404

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d’état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

GetClusterSnapshot

Service : Amazon DocumentDB Elastic Clusters

Renvoie des informations sur un instantané de cluster élastique spécifique

Syntaxe de la demande

```
GET /cluster-snapshot/snapshotArn HTTP/1.1
```

Paramètres de demande URI

La demande utilise les paramètres URI suivants.

snapshotArn

Identifiant ARN de l'instantané du cluster élastique.

Obligatoire : oui

Corps de la demande

La demande n’a pas de corps de requête.

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "snapshot": {
      "adminUserName": "string",
      "clusterArn": "string",
      "clusterCreationTime": "string",
      "kmsKeyId": "string",
      "snapshotArn": "string",
      "snapshotCreationTime": "string",
      "snapshotName": "string",
      "snapshotType": "string",
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
```

```
}
```

Eléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

snapshot

Renvoie des informations sur un instantané de cluster élastique spécifique.

Type : objet ClusterSnapshot

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

AccessDeniedException

Exception qui se produit lorsque les autorisations ne sont pas suffisantes pour effectuer une action.

Code d’état HTTP : 403

InternalServerException

Une erreur interne s'est produite au niveau du serveur.

Code d’état HTTP : 500

ResourceNotFoundException

La ressource spécifiée n'a pas pu être localisée.

Code d’état HTTP : 404

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d’état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

ListClusters

Service : Amazon DocumentDB Elastic Clusters

Renvoie des informations sur les clusters élastiques Amazon DocumentDB provisionnés.

Syntaxe de la demande

```
GET /clusters?maxResults=maxResults&nextToken=nextToken HTTP/1.1
```

Paramètres de demande URI

La demande utilise les paramètres URI suivants.

maxResults

Nombre maximal de résultats de capture instantanée du cluster élastique à recevoir dans la réponse.

Plage valide : valeur minimum de 1. Valeur maximale fixée à 100.

nextToken

Un jeton de pagination fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement les enregistrements situés au-delà de ce jeton, jusqu'à la valeur spécifiée par`max-results`.

S'il n'y a plus de données dans la réponse, elles ne `nextToken` seront pas renvoyées.

Corps de la requête

La demande n’a pas de corps de requête.

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "clusters": [
      {
         "clusterArn": "string",
         "clusterName": "string",
```

```
            "status": "string"
        }
    ],
    "nextToken": "string"
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

clusters

Liste des clusters élastiques Amazon DocumentDB.

Type : tableau d'objets ClusterInList

nextToken

Un jeton de pagination fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement les enregistrements situés au-delà de ce jeton, jusqu'à la valeur spécifiée par`max-results`.

S'il n'y a plus de données dans la réponse, elles ne `nextToken` seront pas renvoyées.

Type : chaîne

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

AccessDeniedException

Exception qui se produit lorsque les autorisations ne sont pas suffisantes pour effectuer une action.

Code d'état HTTP : 403

InternalServerException

Une erreur interne du serveur s'est produite.

Code d’état HTTP : 500

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d’état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

ListClusterSnapshots

Service : Amazon DocumentDB Elastic Clusters

Renvoie des informations sur les instantanés d'un cluster élastique spécifié.

Syntaxe de la demande

```
GET /cluster-snapshots?
clusterArn=clusterArn&maxResults=maxResults&nextToken=nextToken&snapshotType=snapshotType
 HTTP/1.1
```

Paramètres de demande URI

La demande utilise les paramètres URI suivants.

clusterArn

L'identifiant ARN du cluster élastique.

maxResults

Le nombre maximal de résultats de capture instantanée du cluster élastique à recevoir dans la réponse.

Plage valide : valeur minimale de 20. Valeur maximale fixée à 100.

nextToken

Un jeton de pagination fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement les enregistrements situés au-delà de ce jeton, jusqu'à la valeur spécifiée par`max-results`.

S'il n'y a plus de données dans la réponse, elles ne `nextToken` seront pas renvoyées.

snapshotType

Type de snapshots de cluster à renvoyer. Vous pouvez spécifier l'une des valeurs suivantes :

- `automated`- Renvoie tous les instantanés de cluster qu'Amazon DocumentDB a automatiquement créés pour AWS votre compte.
- `manual`- Renvoie tous les instantanés de cluster que vous avez créés manuellement pour votre AWS compte.

Corps de la requête

La demande n’a pas de corps de requête.

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "nextToken": "string",
   "snapshots": [
      {
         "clusterArn": "string",
         "snapshotArn": "string",
         "snapshotCreationTime": "string",
         "snapshotName": "string",
         "status": "string"
      }
   ]
}
```

Eléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

nextToken

Un jeton de pagination fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement les enregistrements situés au-delà de ce jeton, jusqu'à la valeur spécifiée par`max-results`.

S'il n'y a plus de données dans la réponse, elles ne `nextToken` seront pas renvoyées.

Type : chaîne

snapshots

Liste des instantanés pour un cluster élastique spécifié.

Type : tableau d’objets ClusterSnapshotInList

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

AccessDeniedException

Exception qui se produit lorsque les autorisations ne sont pas suffisantes pour effectuer une action.

Code d’état HTTP : 403

InternalServerException

Une erreur interne du serveur s'est produite.

Code d’état HTTP : 500

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d’état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript

- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

ListTagsForResource

Service : Amazon DocumentDB Elastic Clusters

Répertorie toutes les balises d'une ressource de cluster élastique

Syntaxe de la demande

```
GET /tags/resourceArn HTTP/1.1
```

Paramètres de demande URI

La demande utilise les paramètres URI suivants.

resourceArn

Identifiant ARN de la ressource Elastic Cluster.

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 1011.

Obligatoire : oui

Corps de la demande

La demande n’a pas de corps de requête.

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "tags": {
      "string" : "string"
   }
}
```

Eléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

tags

La liste des balises pour la ressource de cluster élastique spécifiée.

Type : mappage chaîne/chaîne

Contraintes de longueur de clé : longueur minimale de 1. Longueur maximale de 128.

Modèle de clé :`^(?!aws:)[a-zA-Z+-=._:/]+$`

Contraintes de longueur de valeur : longueur minimale de 0. Longueur maximale de 256.

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

InternalServerException

Une erreur interne s'est produite au niveau du serveur.

Code d’état HTTP : 500

ResourceNotFoundException

La ressource spécifiée n'a pas pu être localisée.

Code d’état HTTP : 404

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d’état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

RestoreClusterFromSnapshot

Service : Amazon DocumentDB Elastic Clusters

Restaure un cluster élastique à partir d'un instantané.

Syntaxe de la demande

```
POST /cluster-snapshot/snapshotArn/restore HTTP/1.1
Content-type: application/json

{
   "clusterName": "string",
   "kmsKeyId": "string",
   "shardCapacity": number,
   "shardInstanceCount": number,
   "subnetIds": [ "string" ],
   "tags": {
      "string" : "string"
   },
   "vpcSecurityGroupIds": [ "string" ]
}
```

Paramètres de demande URI

La demande utilise les paramètres URI suivants.

snapshotArn

Identifiant ARN de l'instantané du cluster élastique.

Obligatoire : oui

Corps de la demande

Cette demande accepte les données suivantes au format JSON.

clusterName

Nom de l'agrégat élastique.

Type : chaîne

Obligatoire : oui

kmsKeyId

Identifiant de clé KMS à utiliser pour chiffrer le nouveau cluster de clusters élastiques Amazon DocumentDB.

L'identifiant de clé KMS est l'Amazon Resource Name (ARN) de la clé de chiffrement KMS. Si vous créez un cluster en utilisant le même compte Amazon qui possède cette clé de chiffrement KMS, vous pouvez utiliser l'alias de clé KMS au lieu de l'ARN comme clé de chiffrement KMS.

Si aucune clé de chiffrement n'est spécifiée ici, Amazon DocumentDB utilise la clé de chiffrement par défaut créée par KMS pour votre compte. Votre compte possède une clé de chiffrement par défaut différente pour chaque région Amazon.

Type : chaîne

Obligatoire : non

shardCapacity

Capacité de chaque partition du nouveau cluster élastique restauré.

Type : entier

Obligatoire : non

shardInstanceCount

Le nombre d'instances de répliques s'appliquant à toutes les partitions du cluster élastique. Une `shardInstanceCount` valeur de 1 signifie qu'il existe une instance d'écriture et que toutes les instances supplémentaires sont des répliques qui peuvent être utilisées pour les lectures et pour améliorer la disponibilité.

Type : entier

Obligatoire : non

subnetIds

Les identifiants de sous-réseau Amazon EC2 pour le cluster élastique.

Type : tableau de chaînes

Obligatoire : non

tags

Une liste des noms de balises à attribuer au cluster élastique restauré, sous la forme d'un tableau de paires clé-valeur dans lequel la clé est le nom de balise et la valeur est la valeur clé.

Type : mappage chaîne/chaîne

Contraintes de longueur de clé : longueur minimale de 1. Longueur maximale de 128.

Modèle de clé :`^(?!aws:)[a-zA-Z+-=._:/]+$`

Contraintes de longueur de valeur : longueur minimale de 0. Longueur maximale de 256.

Obligatoire : non

vpcSecurityGroupIds

Liste des groupes de sécurité VPC EC2 à associer au cluster élastique.

Type : tableau de chaînes

Obligatoire : non

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
      "shardInstanceCount": number,
```

```
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

cluster

Renvoie des informations sur le cluster élastique restauré.

Type : objet Cluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

AccessDeniedException

Exception qui se produit lorsque les autorisations ne sont pas suffisantes pour effectuer une action.

Code d'état HTTP : 403

ConflictException

Il y a eu un conflit d'accès.

Code d'état HTTP : 409

InternalServerException

Une erreur interne s'est produite au niveau du serveur.

Code d’état HTTP : 500

ResourceNotFoundException

La ressource spécifiée n'a pas pu être localisée.

Code d’état HTTP : 404

ServiceQuotaExceededException

Le quota de service pour l'action a été dépassé.

Code d'état HTTP : 402

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d’état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3

- AWS SDK pour Python
- AWS SDK pour Ruby V3

StartCluster

Service : Amazon DocumentDB Elastic Clusters

Redémarre le cluster élastique arrêté spécifié par`clusterARN`.

Syntaxe de la demande

```
POST /cluster/clusterArn/start HTTP/1.1
```

Paramètres de demande URI

La demande utilise les paramètres URI suivants.

clusterArn

L'identifiant ARN du cluster élastique.

Obligatoire : oui

Corps de la demande

La demande n’a pas de corps de requête.

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
```

```
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

cluster

Renvoie des informations sur un cluster élastique spécifique.

Type : objet Cluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

AccessDeniedException

Exception qui se produit lorsque les autorisations ne sont pas suffisantes pour effectuer une action.

Code d'état HTTP : 403

InternalServerException

Une erreur interne s'est produite au niveau du serveur.

Code d'état HTTP : 500

ResourceNotFoundException

La ressource spécifiée n'a pas pu être localisée.

Code d’état HTTP : 404

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d’état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

StopCluster

Service : Amazon DocumentDB Elastic Clusters

Arrête le cluster élastique en cours d'exécution spécifié par`clusterArn`. Le cluster élastique doit être dans l'état disponible.

Syntaxe de la demande

```
POST /cluster/clusterArn/stop HTTP/1.1
```

Paramètres de demande URI

La demande utilise les paramètres URI suivants.

clusterArn

L'identifiant ARN du cluster élastique.

Obligatoire : oui

Corps de la demande

La demande n’a pas de corps de requête.

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
```

```
      "shardCapacity": number,
      "shardCount": number,
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

cluster

Renvoie des informations sur un cluster élastique spécifique.

Type : objet Cluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

AccessDeniedException

Exception qui se produit lorsque les autorisations ne sont pas suffisantes pour effectuer une action.

Code d'état HTTP : 403

InternalServerException

Une erreur interne s'est produite au niveau du serveur.

Code d’état HTTP : 500

ResourceNotFoundException

La ressource spécifiée n'a pas pu être localisée.

Code d’état HTTP : 404

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d’état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

TagResource

Service : Amazon DocumentDB Elastic Clusters

Ajoute des balises de métadonnées à une ressource de cluster élastique

Syntaxe de la demande

```
POST /tags/resourceArn HTTP/1.1
Content-type: application/json

{
   "tags": {
      "string" : "string"
   }
}
```

Paramètres de demande URI

La demande utilise les paramètres URI suivants.

resourceArn

Identifiant ARN de la ressource Elastic Cluster.

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 1011.

Obligatoire : oui

Corps de la demande

Cette demande accepte les données suivantes au format JSON.

tags

Les balises attribuées à la ressource Elastic Cluster.

Type : mappage chaîne/chaîne

Contraintes de longueur de clé : longueur minimale de 1. Longueur maximale de 128.

Modèle de clé :`^(?!aws:)[a-zA-Z+-=._:/]+$`

Contraintes de longueur de valeur : longueur minimale de 0. Longueur maximale de 256.

Obligatoire : oui

Syntaxe de la réponse

```
HTTP/1.1 200
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

InternalServerException

Une erreur interne du serveur s'est produite.

Code d'état HTTP : 500

ResourceNotFoundException

La ressource spécifiée n'a pas pu être localisée.

Code d'état HTTP : 404

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d'état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

UntagResource

Service : Amazon DocumentDB Elastic Clusters

Supprime les balises de métadonnées d'une ressource de cluster élastique

Syntaxe de la demande

```
DELETE /tags/resourceArn?tagKeys=tagKeys HTTP/1.1
```

Paramètres de demande URI

La demande utilise les paramètres URI suivants.

resourceArn

Identifiant ARN de la ressource Elastic Cluster.

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 1011.

Obligatoire : oui

tagKeys

Les clés de balise à supprimer de la ressource Elastic Cluster.

Membres du tableau : nombre minimum de 0 élément. Nombre maximal de 50 éléments.

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 128.

Modèle : `^(?!aws:)[a-zA-Z+-=._:/]+$`

Obligatoire : oui

Corps de la demande

La demande n’a pas de corps de requête.

Syntaxe de la réponse

```
HTTP/1.1 200
```

Eléments de réponse

Si l’action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

InternalServerException

Une erreur interne du serveur s'est produite.

Code d’état HTTP : 500

ResourceNotFoundException

La ressource spécifiée n'a pas pu être localisée.

Code d’état HTTP : 404

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d’état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d’état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3

- AWS SDK pour Python
- AWS SDK pour Ruby V3

UpdateCluster

Service : Amazon DocumentDB Elastic Clusters

Modifie un cluster élastique. Cela inclut la mise à jour du nom d'utilisateur/mot de passe administrateur, la mise à niveau de la version de l'API et la configuration d'une fenêtre de sauvegarde et d'une fenêtre de maintenance

Syntaxe de la demande

```
PUT /cluster/clusterArn HTTP/1.1
Content-type: application/json

{
   "adminUserPassword": "string",
   "authType": "string",
   "backupRetentionPeriod": number,
   "clientToken": "string",
   "preferredBackupWindow": "string",
   "preferredMaintenanceWindow": "string",
   "shardCapacity": number,
   "shardCount": number,
   "shardInstanceCount": number,
   "subnetIds": [ "string" ],
   "vpcSecurityGroupIds": [ "string" ]
}
```

Paramètres de demande URI

La demande utilise les paramètres URI suivants.

clusterArn

L'identifiant ARN du cluster élastique.

Obligatoire : oui

Corps de la demande

Cette demande accepte les données suivantes au format JSON.

adminUserPassword

Le mot de passe associé à l'administrateur du cluster élastique. Ce mot de passe peut contenir tout caractère ASCII imprimable à l'exception de la barre oblique (/), des guillemets doubles (") ou du symbole arobase (@).

Contraintes : Doit contenir de 8 à 100 caractères.

Type : chaîne

Obligatoire : non

authType

Type d'authentification utilisé pour déterminer où récupérer le mot de passe utilisé pour accéder au cluster élastique. Les types valides sont `PLAIN_TEXT` ou`SECRET_ARN`.

Type : chaîne

Valeurs valides : `PLAIN_TEXT | SECRET_ARN`

Obligatoire : non

backupRetentionPeriod

Nombre de jours pendant lesquels les instantanés automatiques sont conservés.

Type : entier

Obligatoire : non

clientToken

Le jeton client pour le cluster élastique.

Type : chaîne

Obligatoire : non

preferredBackupWindow

La plage horaire quotidienne pendant laquelle les sauvegardes automatisées sont créées si les sauvegardes automatisées sont activées, comme déterminé par le`backupRetentionPeriod`.

Type : chaîne

Obligatoire : non

preferredMaintenanceWindow

Intervalle de temps hebdomadaire, au format Universal Coordinated Time (UTC), pendant lequel a lieu la maintenance du système.

Format : `ddd:hh24:mi-ddd:hh24:mi`

Par défaut : une fenêtre de 30 minutes sélectionnée au hasard dans un intervalle de 8 heures pour chacune d'elles Région AWS, survenant un jour aléatoire de la semaine.

Jours valides : lundi, mardi, mercredi, jeudi, vendredi, samedi, dimanche

Contraintes : fenêtre minimale de 30 minutes.

Type : chaîne

Obligatoire : non

shardCapacity

Le nombre de vCPU assignés à chaque partition de cluster élastique. Le maximum est de 64. Les valeurs autorisées sont 2, 4, 8, 16, 32, 64.

Type : entier

Obligatoire : non

shardCount

Le nombre de partitions attribuées au cluster élastique. Le maximum est de 32.

Type : entier

Obligatoire : non

shardInstanceCount

Le nombre d'instances de répliques s'appliquant à toutes les partitions du cluster élastique. Une `shardInstanceCount` valeur de 1 signifie qu'il existe une instance d'écriture et que toutes les instances supplémentaires sont des répliques qui peuvent être utilisées pour les lectures et pour améliorer la disponibilité.

Type : entier

Obligatoire : non

subnetIds

Les identifiants de sous-réseau Amazon EC2 pour le cluster élastique.

Type : tableau de chaînes

Obligatoire : non

vpcSecurityGroupIds

Liste des groupes de sécurité VPC EC2 à associer au cluster élastique.

Type : tableau de chaînes

Obligatoire : non

Syntaxe de la réponse

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
```

```
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Eléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON par le service.

cluster

Renvoie des informations sur le cluster élastique mis à jour.

Type : objet Cluster

Erreurs

Pour plus d'informations sur les erreurs courantes pour toutes les actions, consultez Erreurs courantes.

AccessDeniedException

Exception qui se produit lorsque les autorisations ne sont pas suffisantes pour effectuer une action.

Code d'état HTTP : 403

ConflictException

Il y a eu un conflit d'accès.

Code d'état HTTP : 409

InternalServerException

Une erreur interne du serveur s'est produite.

Code d'état HTTP : 500

ResourceNotFoundException

La ressource spécifiée n'a pas pu être localisée.

Code d'état HTTP : 404

ThrottlingException

ThrottlingException sera lancé lorsque la demande a été refusée en raison de la limitation des demandes.

Code d'état HTTP : 429

ValidationException

Structure définissant une exception de validation.

Code d'état HTTP : 400

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- Interface de ligne de commande AWS
- AWS SDK pour .NET
- AWS SDK pour C++
- AWS SDK pour Go v2
- AWS SDK pour Java V2
- AWS SDK pour V3 JavaScript
- AWS SDK pour PHP V3
- AWS SDK pour Python
- AWS SDK pour Ruby V3

Types de données

Les types de données suivants sont pris en charge par Amazon DocumentDB (with MongoDB compatibility) :

- AvailabilityZone
- Certificate
- CertificateDetails

- CloudwatchLogsExportConfiguration
- DBCluster
- DBClusterMember
- DBClusterParameterGroup
- DBClusterRole
- DBClusterSnapshot
- DBClusterSnapshotAttribute
- DBClusterSnapshotAttributesResult
- DBEngineVersion
- DBInstance
- DBInstanceStatusInfo
- DBSubnetGroup
- Endpoint
- EngineDefaults
- Event
- EventCategoriesMap
- EventSubscription
- Filter
- GlobalCluster
- GlobalClusterMember
- OrderableDBInstanceOption
- Parameter
- PendingCloudwatchLogsExports
- PendingMaintenanceAction
- PendingModifiedValues
- ResourcePendingMaintenanceActions
- Subnet
- Tag
- UpgradeTarget
- VpcSecurityGroupMembership

Les types de données suivants sont pris en charge par Amazon DocumentDB Elastic Clusters :

- Cluster
- ClusterInList
- ClusterSnapshot
- ClusterSnapshotInList
- Shard
- ValidationExceptionField

Amazon DocumentDB (with MongoDB compatibility)

Les types de données suivants sont pris en charge par Amazon DocumentDB (with MongoDB compatibility) :

- AvailabilityZone
- Certificate
- CertificateDetails
- CloudwatchLogsExportConfiguration
- DBCluster
- DBClusterMember
- DBClusterParameterGroup
- DBClusterRole
- DBClusterSnapshot
- DBClusterSnapshotAttribute
- DBClusterSnapshotAttributesResult
- DBEngineVersion
- DBInstance
- DBInstanceStatusInfo
- DBSubnetGroup
- Endpoint
- EngineDefaults
- Event

- EventCategoriesMap
- EventSubscription
- Filter
- GlobalCluster
- GlobalClusterMember
- OrderableDBInstanceOption
- Parameter
- PendingCloudwatchLogsExports
- PendingMaintenanceAction
- PendingModifiedValues
- ResourcePendingMaintenanceActions
- Subnet
- Tag
- UpgradeTarget
- VpcSecurityGroupMembership

AvailabilityZone

Service : Amazon DocumentDB (with MongoDB compatibility)

Informations sur une zone de disponibilité.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

Name

Nom de la zone de disponibilité.

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Certificate

Service : Amazon DocumentDB (with MongoDB compatibility)

Un certificat d'autorité de certification (CA) pour un Compte AWS.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

CertificateArn

L'Amazon Resource Name (ARN) du certificat.

Exemple : `arn:aws:rds:us-east-1::cert:rds-ca-2019`

Type : chaîne

Obligatoire : non

CertificateIdentifier

La clé unique qui identifie un certificat.

Exemple : `rds-ca-2019`

Type : chaîne

Obligatoire : non

CertificateType

Type du certificat.

Exemple : `CA`

Type : chaîne

Obligatoire : non

Thumbprint

Empreinte numérique du certificat.

Type : chaîne

Obligatoire : non

ValidFrom

Date-heure de début à partir de laquelle le certificat est valide.

Exemple : `2019-07-31T17:57:09Z`

Type : Timestamp

Obligatoire : non

ValidTill

Date-heure après laquelle le certificat n'est plus valide.

Exemple : `2024-07-31T17:57:09Z`

Type : Timestamp

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

CertificateDetails

Service : Amazon DocumentDB (with MongoDB compatibility)

Renvoie les détails du certificat de serveur de l'instance de base de données.

Pour plus d'informations, consultez la section Mise à jour de vos certificats TLS Amazon DocumentDB et chiffrement des données en transit dans le guide du développeur Amazon DocumentDB.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

CAIdentifier

L'identifiant CA du certificat CA utilisé pour le certificat de serveur de l'instance de base de données.

Type : chaîne

Obligatoire : non

ValidTill

Date d'expiration du certificat de serveur de l'instance de base de données.

Type : Timestamp

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

CloudwatchLogsExportConfiguration

Service : Amazon DocumentDB (with MongoDB compatibility)

Le paramètre de configuration des types de journaux à activer pour l'exportation vers Amazon CloudWatch Logs pour une instance ou un cluster spécifique.

Les `DisableLogTypes` tableaux `EnableLogTypes` et déterminent quels journaux sont exportés (ou non exportés) vers CloudWatch Logs. Les valeurs contenues dans ces tableaux dépendent du moteur utilisé.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

DisableLogTypes.member.N

Liste des types de journaux à désactiver.

Type : tableau de chaînes

Obligatoire : non

EnableLogTypes.member.N

Liste des types de journaux à activer.

Type : tableau de chaînes

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DBCluster

Service : Amazon DocumentDB (with MongoDB compatibility)

Informations détaillées sur un cluster.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

AssociatedRoles.DBClusterRole.N

Fournit une liste des rôles AWS Identity and Access Management (IAM) associés au cluster. Les rôles (IAM) associés à un cluster autorisent le cluster à accéder à d'autres AWS services en votre nom.

Type : tableau d’objets DBClusterRole

Obligatoire : non

AvailabilityZones.AvailabilityZone.N

Fournit la liste des zones de disponibilité Amazon EC2 dans lesquelles les instances du cluster peuvent être créées.

Type : tableau de chaînes

Obligatoire : non

BackupRetentionPeriod

Spécifie le nombre de jours pendant lesquels les instantanés automatiques sont conservés.

Type : entier

Obligatoire : non

CloneGroupId

Identifie le groupe de clones auquel est associé le cluster de bases de données.

Type : chaîne

Obligatoire : non

ClusterCreateTime

Spécifie l'heure à laquelle le cluster a été créé, en temps universel coordonné (UTC).

Type : Timestamp

Obligatoire : non

DBClusterArn

Le nom de ressource Amazon (ARN) du cluster.

Type : chaîne

Obligatoire : non

DBClusterIdentifier

Contient un identifiant de cluster fourni par l'utilisateur. Cet identifiant est la clé unique qui identifie un cluster.

Type : chaîne

Obligatoire : non

DBClusterMembers.DBClusterMember.N

Fournit la liste des instances qui constituent le cluster.

Type : tableau d’objets DBClusterMember

Obligatoire : non

DBClusterParameterGroup

Spécifie le nom du groupe de paramètres de cluster pour le cluster.

Type : chaîne

Obligatoire : non

DbClusterResourceId

L'identifiant Région AWS unique et immuable du cluster. Cet identifiant se trouve dans les entrées du AWS CloudTrail journal chaque fois que l'on accède à la AWS KMS clé du cluster.

Type : chaîne

Obligatoire : non

DBSubnetGroup

Spécifie des informations sur le groupe de sous-réseaux associé au cluster, notamment le nom, la description et les sous-réseaux du groupe de sous-réseaux.

Type : chaîne

Obligatoire : non

DeletionProtection

Spécifie si ce cluster peut être supprimé. Si cette option `DeletionProtection` est activée, le cluster ne peut pas être supprimé sauf s'il `DeletionProtection` est modifié et désactivé. `DeletionProtection`protège les clusters contre la suppression accidentelle.

Type : booléen

Obligatoire : non

EarliestRestorableTime

Heure la plus proche à laquelle une base de données peut être restaurée par point-in-time restauration.

Type : Timestamp

Obligatoire : non

EnabledCloudwatchLogsExports.member.N

Liste des types de journaux que ce cluster est configuré pour exporter vers Amazon CloudWatch Logs.

Type : tableau de chaînes

Obligatoire : non

Endpoint

Spécifie le point de terminaison de connexion pour l'instance principale du cluster.

Type : chaîne

Obligatoire : non

Engine

Fournit le nom du moteur de base de données à utiliser pour ce cluster.

Type : chaîne

Obligatoire : non

EngineVersion

Indique la version du moteur de base de données.

Type : chaîne

Obligatoire : non

HostedZoneId

Spécifie l'ID attribué par Amazon Route 53 lorsque vous créez une zone hébergée.

Type : chaîne

Obligatoire : non

KmsKeyId

Si tel `StorageEncrypted` est le cas`true`, l'identifiant de AWS KMS clé du cluster chiffré.

Type : chaîne

Obligatoire : non

LatestRestorableTime

Spécifie l'heure limite à laquelle une base de données peut être restaurée à l'aide de la fonction de point-in-time restauration.

Type : Timestamp

Obligatoire : non

MasterUsername

Contient le nom d'utilisateur principal du cluster.

Type : chaîne

Obligatoire : non

MultiAZ

Spécifie si le cluster possède des instances dans plusieurs zones de disponibilité.

Type : booléen

Obligatoire : non

PercentProgress

Spécifie la progression de l'opération sous forme de pourcentage.

Type : chaîne

Obligatoire : non

Port

Spécifie le port sur lequel le moteur de base de données est à l'écoute.

Type : entier

Obligatoire : non

PreferredBackupWindow

Spécifie la plage de temps quotidienne au cours de laquelle des sauvegardes automatiques sont créées si cette fonctionnalité est activée, comme déterminé par la propriété `BackupRetentionPeriod`.

Type : chaîne

Obligatoire : non

PreferredMaintenanceWindow

Spécifie l'intervalle de temps hebdomadaire, au format Universal Coordinated Time (UTC), pendant lequel a lieu la maintenance du système.

Type : chaîne

Obligatoire : non

ReaderEndpoint

Point de terminaison du lecteur pour le cluster. Le point de terminaison du lecteur d'un cluster équilibre la charge des connexions entre les répliques Amazon DocumentDB disponibles dans un cluster. Lorsque les clients demandent de nouvelles connexions au point de terminaison du lecteur, Amazon DocumentDB distribue les demandes de connexion entre les répliques Amazon DocumentDB du cluster. Cette fonctionnalité permet d'équilibrer votre charge de travail de lecture sur plusieurs répliques Amazon DocumentDB de votre cluster.

Si un basculement se produit et que la réplique Amazon DocumentDB à laquelle vous êtes connecté est promue instance principale, votre connexion est interrompue. Pour continuer à envoyer votre charge de travail de lecture à d'autres répliques Amazon DocumentDB du cluster, vous pouvez ensuite vous reconnecter au point de terminaison du lecteur.

Type : chaîne

Obligatoire : non

ReadReplicaIdentifiers.ReadReplicaIdentifier.N

Contient un ou plusieurs identifiants des clusters secondaires associés à ce cluster.

Type : tableau de chaînes

Obligatoire : non

ReplicationSourceIdentifier

Contient l'identifiant du cluster source s'il s'agit d'un cluster secondaire.

Type : chaîne

Obligatoire : non

Status

Spécifie l'état actuel de ce cluster.

Type : chaîne

Obligatoire : non

StorageEncrypted

Indique si le cluster est chiffré.

Type : booléen

Obligatoire : non

StorageType

Type de stockage associé à votre cluster

Type de stockage associé à votre cluster

Pour plus d'informations sur les types de stockage pour les clusters Amazon DocumentDB, consultez la section Configurations de stockage des clusters dans le manuel Amazon DocumentDB Developer Guide.

Valeurs valides pour le type de stockage - `standard | iopt1`

La valeur par défaut est `standard`

Type : chaîne

Obligatoire : non

VpcSecurityGroups.VpcSecurityGroupMembership.N

Fournit une liste des groupes de sécurité du cloud privé virtuel (VPC) auxquels appartient le cluster.

Type : tableau d'objets VpcSecurityGroupMembership

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DBClusterMember

Service : Amazon DocumentDB (with MongoDB compatibility)

Contient des informations sur une instance faisant partie d'un cluster.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

DBClusterParameterGroupStatus

Spécifie l'état du groupe de paramètres de cluster pour ce membre du cluster de base de données.

Type : chaîne

Obligatoire : non

DBInstanceIdentifier

Spécifie l'identifiant d'instance pour ce membre du cluster.

Type : chaîne

Obligatoire : non

IsClusterWriter

Une valeur indiquant `true` si le membre du cluster est l'instance principale du cluster `false` ou non.

Type : booléen

Obligatoire : non

PromotionTier

Valeur qui indique l'ordre dans lequel une réplique Amazon DocumentDB est promue vers l'instance principale après une défaillance de l'instance principale existante.

Type : entier

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DBClusterParameterGroup

Service : Amazon DocumentDB (with MongoDB compatibility)

Informations détaillées sur un groupe de paramètres de cluster.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

DBClusterParameterGroupArn

Amazon Resource Name (ARN) pour le groupe de paramètres du cluster.

Type : chaîne

Obligatoire : non

DBClusterParameterGroupName

Fournit le nom du groupe de paramètres du cluster.

Type : chaîne

Obligatoire : non

DBParameterGroupFamily

Fournit le nom de la famille de groupes de paramètres avec laquelle ce groupe de paramètres de cluster est compatible.

Type : chaîne

Obligatoire : non

Description

Fournit la description spécifiée par le client pour ce groupe de paramètres de cluster.

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DBClusterRole

Service : Amazon DocumentDB (with MongoDB compatibility)

Décrit un rôle AWS Identity and Access Management (IAM) associé à un cluster.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

RoleArn

Le nom de ressource Amazon (ARN) de l'IAMRole associé au cluster de base de données.

Type : chaîne

Obligatoire : non

Status

Décrit l'état de l'association entre l'IAMRole et le cluster. La `Status` propriété renvoie l'une des valeurs suivantes :

- `ACTIVE`- L'ARN IAMRole est associé au cluster et peut être utilisé pour accéder à d'autres AWS services en votre nom.
- `PENDING`- L'ARN IAMRole est associé au cluster.
- `INVALID`- L'ARN IAMRole est associé au cluster, mais celui-ci ne peut pas assumer le IAMRole pour accéder à d'autres AWS services en votre nom.

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++

- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DBClusterSnapshot

Service : Amazon DocumentDB (with MongoDB compatibility)

Informations détaillées sur un instantané de cluster.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

AvailabilityZones.AvailabilityZone.N

Fournit la liste des zones de disponibilité Amazon EC2 dans lesquelles les instances du snapshot du cluster peuvent être restaurées.

Type : tableau de chaînes

Obligatoire : non

ClusterCreateTime

Spécifie l'heure à laquelle le cluster a été créé, en temps universel coordonné (UTC).

Type : Timestamp

Obligatoire : non

DBClusterIdentifier

Spécifie l'identifiant du cluster à partir duquel ce cliché de cluster a été créé.

Type : chaîne

Obligatoire : non

DBClusterSnapshotArn

Le nom de ressource Amazon (ARN) pour l'instantané du cluster.

Type : chaîne

Obligatoire : non

DBClusterSnapshotIdentifier

Spécifie l'identifiant du cliché du cluster.

Type : chaîne

Obligatoire : non

Engine

Spécifie le nom du moteur de base de données.

Type : chaîne

Obligatoire : non

EngineVersion

Fournit la version du moteur de base de données pour cet instantané de cluster.

Type : chaîne

Obligatoire : non

KmsKeyId

Si tel `StorageEncrypted` est le cas`true`, l'identifiant de AWS KMS clé pour le snapshot du cluster chiffré.

Type : chaîne

Obligatoire : non

MasterUsername

Fournit le nom d'utilisateur principal pour le snapshot du cluster.

Type : chaîne

Obligatoire : non

PercentProgress

Spécifie une estimation du pourcentage de données transférées.

Type : entier

Obligatoire : non

Port

Spécifie le port sur lequel le cluster écoutait au moment de la capture instantanée.

Type : entier

Obligatoire : non

SnapshotCreateTime

Indique l'heure à laquelle le cliché a été pris, en UTC.

Type : Timestamp

Obligatoire : non

SnapshotType

Fournit le type de capture instantanée du cluster.

Type : chaîne

Obligatoire : non

SourceDBClusterSnapshotArn

Si le cliché de cluster a été copié à partir d'un instantané de cluster source, l'ARN de l'instantané de cluster source ; sinon, une valeur nulle.

Type : chaîne

Obligatoire : non

Status

Spécifie l'état de ce cliché de cluster.

Type : chaîne

Obligatoire : non

StorageEncrypted

Spécifie si le snapshot du cluster est chiffré.

Type : booléen

Obligatoire : non

StorageType

Type de stockage associé à votre instantané de cluster

Pour plus d'informations sur les types de stockage pour les clusters Amazon DocumentDB, consultez la section Configurations de stockage des clusters dans le manuel Amazon DocumentDB Developer Guide.

Valeurs valides pour le type de stockage - `standard | iopt1`

La valeur par défaut est `standard`

Type : chaîne

Obligatoire : non

VpcId

Fournit l'ID de cloud privé virtuel (VPC) associé au snapshot du cluster.

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DBClusterSnapshotAttribute

Service : Amazon DocumentDB (with MongoDB compatibility)

Contient le nom et les valeurs d'un attribut de capture d'écran de cluster manuel.

Les attributs de capture d'écran de cluster manuels sont utilisés pour autoriser d'autres personnes Comptes AWS à restaurer un instantané de cluster manuel.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

AttributeName

Nom de l'attribut de capture d'écran manuel du cluster.

L'attribut nommé `restore` fait référence à la liste Comptes AWS des personnes autorisées à copier ou à restaurer l'instantané manuel du cluster.

Type : chaîne

Obligatoire : non

AttributeValues.AttributeValue.N

Les valeurs de l'attribut de capture d'écran manuel du cluster.

Si le `AttributeName` champ est défini sur`restore`, cet élément renvoie une liste des Comptes AWS identifiants autorisés à copier ou à restaurer l'instantané manuel du cluster. Si la valeur de `all` figure dans la liste, l'instantané manuel du cluster est public et peut être copié ou restauré par tout Compte AWS le monde.

Type : tableau de chaînes

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DBClusterSnapshotAttributesResult

Service : Amazon DocumentDB (with MongoDB compatibility)

Informations détaillées sur les attributs associés à un instantané de cluster.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

DBClusterSnapshotAttributes.DBClusterSnapshotAttribute.N

La liste des attributs et des valeurs de l'instantané du cluster.

Type : tableau d’objets DBClusterSnapshotAttribute

Obligatoire : non

DBClusterSnapshotIdentifier

Identifiant du snapshot du cluster auquel les attributs s'appliquent.

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DBEngineVersion

Service : Amazon DocumentDB (with MongoDB compatibility)

Informations détaillées sur une version du moteur.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

DBEngineDescription

Description du moteur de base de données.

Type : chaîne

Obligatoire : non

DBEngineVersionDescription

Description de la version du moteur de base de données.

Type : chaîne

Obligatoire : non

DBParameterGroupFamily

Nom de la famille de groupes de paramètres pour le moteur de base de données.

Type : chaîne

Obligatoire : non

Engine

Nom du moteur de base de données.

Type : chaîne

Obligatoire : non

EngineVersion

Le numéro de version du moteur de base de données.

Type : chaîne

Obligatoire : non

ExportableLogTypes.member.N

Les types de journaux que le moteur de base de données peut exporter vers Amazon CloudWatch Logs.

Type : tableau de chaînes

Obligatoire : non

SupportedCACertificateIdentifiers.member.N

Liste des identifiants de certificats CA pris en charge.

Pour plus d'informations, consultez la section Mise à jour de vos certificats TLS Amazon DocumentDB et chiffrement des données en transit dans le guide du développeur Amazon DocumentDB.

Type : tableau de chaînes

Obligatoire : non

SupportsCertificateRotationWithoutRestart

Indique si la version du moteur prend en charge la rotation du certificat de serveur sans redémarrer l'instance de base de données.

Type : booléen

Obligatoire : non

SupportsLogExportsToCloudwatchLogs

Une valeur qui indique si la version du moteur prend en charge l'exportation des types de journaux spécifiés par `ExportableLogTypes` to CloudWatch Logs.

Type : booléen

Obligatoire : non

ValidUpgradeTarget.UpgradeTarget.N

Liste des versions de moteur vers lesquelles cette version du moteur de base de données peut être mise à niveau.

Type : tableau d'objets UpgradeTarget

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DBInstance

Service : Amazon DocumentDB (with MongoDB compatibility)

Informations détaillées sur une instance.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

AutoMinorVersionUpgrade

Ne s'applique pas. Ce paramètre ne s'applique pas à Amazon DocumentDB. Amazon DocumentDB n'effectue pas de mises à niveau mineures de version, quelle que soit la valeur définie.

Type : booléen

Obligatoire : non

AvailabilityZone

Spécifie le nom de la zone de disponibilité dans laquelle se trouve l'instance.

Type : chaîne

Obligatoire : non

BackupRetentionPeriod

Spécifie le nombre de jours pendant lesquels les instantanés automatiques sont conservés.

Type : entier

Obligatoire : non

CACertificateIdentifier

Identifiant du certificat CA de cette instance de base de données.

Type : chaîne

Obligatoire : non

CertificateDetails

Les détails du certificat de serveur de l'instance de base de données.

Type : objet CertificateDetails

Obligatoire : non

CopyTagsToSnapshot

Une valeur qui indique si vous voulez copier toutes les balises à partir de l'instance de base de données pour les instantanés de l'instance de base de données. Par défaut, les balises ne sont pas copiées.

Type : booléen

Obligatoire : non

DBClusterIdentifier

Contient le nom du cluster dont l'instance est membre si l'instance est membre d'un cluster.

Type : chaîne

Obligatoire : non

DBInstanceArn

L'Amazon Resource Name (ARN) de l'instance.

Type : chaîne

Obligatoire : non

DBInstanceClass

Contient le nom de la classe de capacité de calcul et de mémoire de l'instance.

Type : chaîne

Obligatoire : non

DBInstanceIdentifier

Contient un identifiant de base de données fourni par l'utilisateur. Cet identifiant est la clé unique qui identifie une instance.

Type : chaîne

Obligatoire : non

DBInstanceStatus

Indique l'état actuel de cette base de données.

Type : chaîne

Obligatoire : non

DbiResourceId

L'identifiant Région AWS unique et immuable de l'instance. Cet identifiant se trouve dans les entrées du AWS CloudTrail journal chaque fois que l'on accède à la AWS KMS clé de l'instance.

Type : chaîne

Obligatoire : non

DBSubnetGroup

Spécifie des informations sur le groupe de sous-réseaux associé à l'instance, notamment le nom, la description et les sous-réseaux du groupe de sous-réseaux.

Type : objet DBSubnetGroup

Obligatoire : non

EnabledCloudwatchLogsExports.member.N

Liste des types de journaux que cette instance est configurée pour exporter vers CloudWatch Logs.

Type : tableau de chaînes

Obligatoire : non

Endpoint

Spécifie le point de terminaison de la connexion.

Type : objet Endpoint

Obligatoire : non

Engine

Fournit le nom du moteur de base de données à utiliser pour cette instance.

Type : chaîne

Obligatoire : non

EngineVersion

Indique la version du moteur de base de données.

Type : chaîne

Obligatoire : non

InstanceCreateTime

Indique la date et l'heure de création de l'instance.

Type : Timestamp

Obligatoire : non

KmsKeyId

Si tel `StorageEncrypted` est le cas`true`, l'identifiant de AWS KMS clé de l'instance chiffrée.

Type : chaîne

Obligatoire : non

LatestRestorableTime

Spécifie l'heure limite à laquelle une base de données peut être restaurée par point-in-time restauration.

Type : Timestamp

Obligatoire : non

PendingModifiedValues

Spécifie que les modifications apportées à l'instance sont en attente. Cet élément est inclus uniquement lorsque des modifications sont en attente. Des modifications spécifiques sont identifiées par sous-éléments.

Type : objet PendingModifiedValues

Obligatoire : non

PreferredBackupWindow

Spécifie la plage de temps quotidienne au cours de laquelle des sauvegardes automatiques sont créées si cette fonctionnalité est activée, comme déterminé par la propriété `BackupRetentionPeriod`.

Type : chaîne

Obligatoire : non

PreferredMaintenanceWindow

Spécifie l'intervalle de temps hebdomadaire, au format Universal Coordinated Time (UTC), pendant lequel a lieu la maintenance du système.

Type : chaîne

Obligatoire : non

PromotionTier

Valeur qui indique l'ordre dans lequel une réplique Amazon DocumentDB est promue vers l'instance principale après une défaillance de l'instance principale existante.

Type : entier

Obligatoire : non

PubliclyAccessible

Non pris en charge. Amazon DocumentDB ne prend actuellement pas en charge les points de terminaison publics. La valeur de `PubliclyAccessible` est toujours`false`.

Type : booléen

Obligatoire : non

StatusInfos.DBInstanceStatusInfo.N

État d'une réplique lue. Si l'instance n'est pas une réplique lue, ce champ est vide.

Type : tableau d’objets DBInstanceStatusInfo

Obligatoire : non

StorageEncrypted

Spécifie si l'instance est chiffrée ou non.

Type : booléen

Obligatoire : non

VpcSecurityGroups.VpcSecurityGroupMembership.N

Fournit une liste des éléments du groupe de sécurité VPC auxquels appartient l'instance.

Type : tableau d’objets VpcSecurityGroupMembership

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DBInstanceStatusInfo

Service : Amazon DocumentDB (with MongoDB compatibility)

Fournit une liste d'informations d'état pour une instance.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

Message

Détails de l'erreur en cas d'erreur de l'instance. Si l'instance n'est pas dans un état d'erreur, cette valeur est vide.

Type : chaîne

Obligatoire : non

Normal

Une valeur booléenne indiquant `true` si l'instance fonctionne normalement ou `false` si l'instance est en état d'erreur.

Type : booléen

Obligatoire : non

Status

État de l'instance. Pour une `StatusType` réplique en lecture, les valeurs peuvent être`replicating`, error`stopped`, `outerminated`.

Type : chaîne

Obligatoire : non

StatusType

Cette valeur est actuellement « »`read replication`.

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

DBSubnetGroup

Service : Amazon DocumentDB (with MongoDB compatibility)

Informations détaillées sur un groupe de sous-réseaux.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

DBSubnetGroupArn

Amazon Resource Name (ARN) du groupe de sous-réseaux de base de données.

Type : chaîne

Obligatoire : non

DBSubnetGroupDescription

Fournit la description du groupe de sous-réseaux.

Type : chaîne

Obligatoire : non

DBSubnetGroupName

Le nom du groupe de sous-réseau.

Type : chaîne

Obligatoire : non

SubnetGroupStatus

Indique le statut du groupe de sous-réseaux.

Type : chaîne

Obligatoire : non

Subnets.Subnet.N

Informations détaillées sur un ou plusieurs sous-réseaux au sein d'un groupe de sous-réseaux.

Type : tableau d’objets Subnet

Obligatoire : non

VpcId

Fournit l'ID de cloud privé virtuel (VPC) du groupe de sous-réseaux.

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Endpoint

Service : Amazon DocumentDB (with MongoDB compatibility)

Informations réseau pour accéder à un cluster ou à une instance. Les programmes clients doivent spécifier un point de terminaison valide pour accéder à ces ressources Amazon DocumentDB.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

Address

Spécifie l'adresse DNS de l'instance.

Type : chaîne

Obligatoire : non

HostedZoneId

Spécifie l'ID attribué par Amazon Route 53 lorsque vous créez une zone hébergée.

Type : chaîne

Obligatoire : non

Port

Spécifie le port sur lequel le moteur de base de données est à l'écoute.

Type : entier

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++

- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

EngineDefaults

Service : Amazon DocumentDB (with MongoDB compatibility)

Contient le résultat d'une invocation réussie de l'`DescribeEngineDefaultClusterParameters`opération.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

DBParameterGroupFamily

Nom de la famille de groupes de paramètres du cluster pour laquelle les informations sur les paramètres du moteur doivent être renvoyées.

Type : chaîne

Obligatoire : non

Marker

Jeton de pagination facultatif fourni par une demande précédente. Si ce paramètre est spécifié, la réponse inclut uniquement des enregistrements supérieurs au marqueur, jusqu'à la valeur spécifiée par `MaxRecords`.

Type : chaîne

Obligatoire : non

Parameters.Parameter.N

Les paramètres d'une famille de groupes de paramètres de cluster particulière.

Type : tableau d’objets Parameter

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Event

Service : Amazon DocumentDB (with MongoDB compatibility)

Informations détaillées sur un événement.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

Date

Spécifie la date et l'heure de l'événement.

Type : Timestamp

Obligatoire : non

EventCategories.EventCategory.N

Spécifie la catégorie pour l'événement.

Type : tableau de chaînes

Obligatoire : non

Message

Fournit le texte de cet événement.

Type : chaîne

Obligatoire : non

SourceArn

ARN (Amazon Resource Name) de l'événement.

Type : chaîne

Obligatoire : non

SourceIdentifier

Fournit l'identifiant de la source de l'événement.

Type : chaîne

Obligatoire : non

SourceType

Spécifie le type de source pour cet événement.

Type : chaîne

Valeurs valides : `db-instance | db-parameter-group | db-security-group | db-snapshot | db-cluster | db-cluster-snapshot`

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

EventCategoriesMap

Service : Amazon DocumentDB (with MongoDB compatibility)

Type de source d'événement, accompagné d'un ou de plusieurs noms de catégories d'événements.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

EventCategories.EventCategory.N

Les catégories d'événements pour le type de source spécifié.

Type : tableau de chaînes

Obligatoire : non

SourceType

Type de source auquel appartiennent les catégories renvoyées.

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

EventSubscription

Service : Amazon DocumentDB (with MongoDB compatibility)

Informations détaillées sur un événement auquel vous vous êtes inscrit.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

CustomerAwsId

Le compte AWS client associé à l'abonnement aux notifications d'événements Amazon DocumentDB.

Type : chaîne

Obligatoire : non

CustSubscriptionId

L'ID d'abonnement aux notifications d'événements Amazon DocumentDB.

Type : chaîne

Obligatoire : non

Enabled

Valeur booléenne indiquant si l'abonnement est activé. La valeur de `true` indique que l'abonnement est activé.

Type : booléen

Obligatoire : non

EventCategoriesList.EventCategory.N

Liste des catégories d'événements pour l'abonnement aux notifications d'événements Amazon DocumentDB.

Type : tableau de chaînes

Obligatoire : non

EventSubscriptionArn

Amazon Resource Name (ARN) de l'abonnement aux événements.

Type : chaîne

Obligatoire : non

SnsTopicArn

L'ARN du sujet de l'abonnement aux notifications d'événements Amazon DocumentDB.

Type : chaîne

Obligatoire : non

SourceIdsList.SourceId.N

Liste des identifiants sources pour l'abonnement aux notifications d'événements Amazon DocumentDB.

Type : tableau de chaînes

Obligatoire : non

SourceType

Type de source pour l'abonnement aux notifications d'événements Amazon DocumentDB.

Type : chaîne

Obligatoire : non

Status

État de l'abonnement aux notifications d'événements Amazon DocumentDB.

Contraintes :

Il peut s'agir de l'une des options suivantes : `creating` `modifyingdeleting`,,`active`,`no-permission`, `topic-not-exist`

Le `no-permission` statut indique qu'Amazon DocumentDB n'est plus autorisé à publier sur le sujet SNS. Le `topic-not-exist` statut indique que le sujet a été supprimé après la création de l'abonnement.

Type : chaîne

Obligatoire : non

SubscriptionCreationTime

Heure à laquelle l'abonnement aux notifications d'événements Amazon DocumentDB a été créé.

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Filter

Service : Amazon DocumentDB (with MongoDB compatibility)

Ensemble nommé de valeurs de filtre, utilisé pour renvoyer une liste de résultats plus spécifique. Vous pouvez utiliser un filtre pour faire correspondre un ensemble de ressources selon des critères spécifiques, tels que les identifiants.

Les caractères génériques ne sont pas pris en charge dans les filtres.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

Name

Nom du filtre. Les noms des filtres distinguent les majuscules et minuscules.

Type : chaîne

Obligatoire : oui

Values.Value.N

Une ou plusieurs valeurs de filtre. Les valeurs de filtre sont sensibles à la casse.

Type : tableau de chaînes

Obligatoire : oui

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

GlobalCluster

Service : Amazon DocumentDB (with MongoDB compatibility)

Type de données représentant un cluster global Amazon DocumentDB.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

DatabaseName

Nom de base de données par défaut au sein du nouveau cluster global.

Type : chaîne

Obligatoire : non

DeletionProtection

Le paramètre de protection contre la suppression pour le nouveau cluster global.

Type : booléen

Obligatoire : non

Engine

Le moteur de base de données Amazon DocumentDB utilisé par le cluster mondial.

Type : chaîne

Obligatoire : non

EngineVersion

Indique la version du moteur de base de données.

Type : chaîne

Obligatoire : non

GlobalClusterArn

Le nom de ressource Amazon (ARN) du cluster mondial.

Type : chaîne

Obligatoire : non

GlobalClusterIdentifier

Contient un identifiant de cluster global fourni par l'utilisateur. Cet identifiant est la clé unique qui identifie un cluster mondial.

Type : chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 255.

Modèle : `[A-Za-z][0-9A-Za-z-:._]*`

Obligatoire : non

GlobalClusterMembers.GlobalClusterMember.N

La liste des identifiants de cluster pour les clusters secondaires au sein du cluster global. Limité à un article pour le moment.

Type : tableau d'objets GlobalClusterMember

Obligatoire : non

GlobalClusterResourceId

Identifiant immuable Région AWS unique du cluster de bases de données global. Cet identifiant se trouve dans les entrées du AWS CloudTrail journal chaque fois que l'on accède à la clé principale du AWS KMS client (CMK) du cluster.

Type : chaîne

Obligatoire : non

Status

Spécifie l'état actuel de ce cluster global.

Type : chaîne

Obligatoire : non

StorageEncrypted

Le paramètre de chiffrement du stockage pour le cluster global.

Type : booléen

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

GlobalClusterMember

Service : Amazon DocumentDB (with MongoDB compatibility)

Structure de données contenant des informations sur les clusters principaux et secondaires associés à un cluster global Amazon DocumentDB.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

DBClusterArn

Le nom de ressource Amazon (ARN) pour chaque cluster Amazon DocumentDB.

Type : chaîne

Obligatoire : non

IsWriter

Spécifie si le cluster Amazon DocumentDB est le cluster principal (c'est-à-dire s'il possède une capacité de lecture-écriture) du cluster global Amazon DocumentDB auquel il est associé.

Type : booléen

Obligatoire : non

Readers.member.N

Le nom de ressource Amazon (ARN) pour chaque cluster secondaire en lecture seule associé au cluster global Aurora.

Type : tableau de chaînes

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

OrderableDBInstanceOption

Service : Amazon DocumentDB (with MongoDB compatibility)

Les options disponibles pour une instance.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

AvailabilityZones.AvailabilityZone.N

Liste des zones de disponibilité pour une instance.

Type : tableau d'objets AvailabilityZone

Obligatoire : non

DBInstanceClass

La classe d'instance d'une instance.

Type : chaîne

Obligatoire : non

Engine

Type de moteur d'une instance.

Type : chaîne

Obligatoire : non

EngineVersion

Version du moteur d'une instance.

Type : chaîne

Obligatoire : non

LicenseModel

Modèle de licence pour une instance.

Type : chaîne

Obligatoire : non

Vpc

Indique si une instance se trouve dans un cloud privé virtuel (VPC).

Type : booléen

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Parameter

Service : Amazon DocumentDB (with MongoDB compatibility)

Informations détaillées sur un paramètre individuel.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

AllowedValues

Spécifie la plage de valeurs valide pour le paramètre.

Type : chaîne

Obligatoire : non

ApplyMethod

Indique quand appliquer les mises à jour de paramètres.

Type : chaîne

Valeurs valides : `immediate | pending-reboot`

Obligatoire : non

ApplyType

Spécifie le type de paramètres spécifiques au moteur.

Type : chaîne

Obligatoire : non

DataType

Spécifie le type de données valide pour le paramètre.

Type : chaîne

Obligatoire : non

Description

Fournit une description du paramètre.

Type : chaîne

Obligatoire : non

IsModifiable

Indique si le paramètre peut être (`true`) ou non (`false`) modifié. Certains paramètres ont des implications en terme de sécurité ou de fonctionnement qui les empêchent d'être modifiés.

Type : booléen

Obligatoire : non

MinimumEngineVersion

Première version de moteur à laquelle le paramètre peut s'appliquer.

Type : chaîne

Obligatoire : non

ParameterName

Spécifie le nom du paramètre.

Type : chaîne

Obligatoire : non

ParameterValue

Spécifie la valeur du paramètre.

Type : chaîne

Obligatoire : non

Source

Indique la source de la valeur du paramètre.

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

PendingCloudwatchLogsExports

Service : Amazon DocumentDB (with MongoDB compatibility)

Liste des types de journaux dont la configuration est toujours en attente. Ces types de journaux sont en cours d'activation ou de désactivation.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

LogTypesToDisable.member.N

Types de journaux en cours d'activation. Une fois activés, ces types de journaux sont exportés vers Amazon CloudWatch Logs.

Type : tableau de chaînes

Obligatoire : non

LogTypesToEnable.member.N

Types de journaux en cours de désactivation. Une fois désactivés, ces types de journaux ne sont pas exportés vers CloudWatch Logs.

Type : tableau de chaînes

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

PendingMaintenanceAction

Service : Amazon DocumentDB (with MongoDB compatibility)

Fournit des informations sur une action de maintenance en attente pour une ressource.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

Action

Type d'action de maintenance en attente disponible pour la ressource.

Type : chaîne

Obligatoire : non

AutoAppliedAfterDate

Date de la fenêtre de maintenance lorsque l'action est appliquée. L'action de maintenance est appliquée à la ressource lors de sa première fenêtre de maintenance après cette date. Si cette date est spécifiée, toutes les demandes de confirmation de l'acceptation `next-maintenance` sont ignorées.

Type : Timestamp

Obligatoire : non

CurrentApplyDate

Date effective d'application de l'action de maintenance en attente à la ressource.

Type : Timestamp

Obligatoire : non

Description

Description fournissant plus de détails sur l'action de maintenance.

Type : chaîne

Obligatoire : non

ForcedApplyDate

Date à laquelle l'action de maintenance est automatiquement appliquée. L'action de maintenance est appliquée à la ressource à cette date indépendamment de la fenêtre de maintenance de la ressource. Si cette date est spécifiée, toutes les demandes de confirmation de l'acceptation `immediate` sont ignorées.

Type : Timestamp

Obligatoire : non

OptInStatus

Indique le type de demande de confirmation de l'acceptation reçue pour la ressource.

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

PendingModifiedValues

Service : Amazon DocumentDB (with MongoDB compatibility)

Un ou plusieurs paramètres modifiés pour une instance. Ces paramètres modifiés ont été demandés, mais n'ont pas encore été appliqués.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

AllocatedStorage

Contient la nouvelle `AllocatedStorage` taille pour l'instance qui sera appliquée ou qui est actuellement appliquée.

Type : entier

Obligatoire : non

BackupRetentionPeriod

Spécifie le nombre de jours en attente pour lesquels des sauvegardes automatiques sont conservées.

Type : entier

Obligatoire : non

CACertificateIdentifier

Spécifie l'identifiant du certificat de l'autorité de certification (CA) pour l'instance de base de données.

Type : chaîne

Obligatoire : non

DBInstanceClass

Contient le nouveau `DBInstanceClass` pour l'instance qui sera appliquée ou qui est actuellement appliquée.

Type : chaîne

Obligatoire : non

DBInstanceIdentifier

Contient le nouveau `DBInstanceIdentifier` pour l'instance qui sera appliquée ou qui est actuellement appliquée.

Type : chaîne

Obligatoire : non

DBSubnetGroupName

Le nouveau groupe de sous-réseaux pour l'instance.

Type : chaîne

Obligatoire : non

EngineVersion

Indique la version du moteur de base de données.

Type : chaîne

Obligatoire : non

Iops

Spécifie la nouvelle valeur d'IOPS provisionnées pour l'instance qui sera appliquée ou qui est actuellement appliquée.

Type : entier

Obligatoire : non

LicenseModel

Modèle de licence pour l'instance.

Valeurs valides: `license-included`, `bring-your-own-license`, `general-public-license`

Type : chaîne

Obligatoire : non

MasterUserPassword

Contient la modification en attente ou en cours des informations d'identification principales pour l'instance.

Type : chaîne

Obligatoire : non

MultiAZ

Indique que l'instance mono-AZ doit passer à un déploiement multi-AZ.

Type : booléen

Obligatoire : non

PendingCloudwatchLogsExports

Liste des types de journaux dont la configuration est toujours en attente. Ces types de journaux sont en cours d'activation ou de désactivation.

Type : objet PendingCloudwatchLogsExports

Obligatoire : non

Port

Spécifie le port en attente pour l'instance.

Type : entier

Obligatoire : non

StorageType

Spécifie le type de stockage à associer à l'instance.

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

ResourcePendingMaintenanceActions

Service : Amazon DocumentDB (with MongoDB compatibility)

Représente la sortie deApplyPendingMaintenanceAction.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

PendingMaintenanceActionDetails.PendingMaintenanceAction.N

Liste qui fournit des détails sur les actions de maintenance en attente pour la ressource.

Type : tableau d'objets PendingMaintenanceAction

Obligatoire : non

ResourceIdentifier

Le nom de ressource Amazon (ARN) de la ressource pour laquelle des actions de maintenance sont en attente.

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Subnet

Service : Amazon DocumentDB (with MongoDB compatibility)

Informations détaillées sur un sous-réseau.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

SubnetAvailabilityZone

Spécifie la zone de disponibilité pour le sous-réseau.

Type : objet AvailabilityZone

Obligatoire : non

SubnetIdentifier

Spécifie l'identifiant du sous-réseau.

Type : chaîne

Obligatoire : non

SubnetStatus

Spécifie le statut du sous-réseau.

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2

- AWS SDK pour Ruby V3

Tag

Service : Amazon DocumentDB (with MongoDB compatibility)

Métadonnées attribuées à une ressource Amazon DocumentDB composée d'une paire clé-valeur.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

Key

Le nom obligatoire de la balise. La valeur de la chaîne peut comporter de 1 à 128 caractères Unicode et ne peut pas être préfixée par « `aws:` » ou « `rds:` ». La chaîne ne peut contenir que l'ensemble des lettres Unicode, des chiffres, des espaces blancs, « _ », « » . ','/',' = ',' + ',' - '(expression régulière Java : « ^ ([\ \ p {L} \ \ p {Z} \ \ p {N} _. : /=+ \ \ -] *) $ »).

Type : chaîne

Obligatoire : non

Value

La valeur facultative de la balise. La valeur de la chaîne peut comporter de 1 à 256 caractères Unicode et ne peut pas être préfixée par « `aws:` » ou « `rds:` ». La chaîne ne peut contenir que l'ensemble des lettres Unicode, des chiffres, des espaces blancs, « _ », « » . ','/',' = ',' + ',' - '(expression régulière Java : « ^ ([\ \ p {L} \ \ p {Z} \ \ p {N} _. : /=+ \ \ -] *) $ »).

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2

- AWS SDK pour Ruby V3

UpgradeTarget

Service : Amazon DocumentDB (with MongoDB compatibility)

Version du moteur de base de données vers laquelle une instance peut être mise à niveau.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

AutoUpgrade

Une valeur qui indique si la version cible est appliquée à toutes les instances de base de données source `AutoMinorVersionUpgrade` définies sur`true`.

Type : booléen

Obligatoire : non

Description

Version du moteur de base de données vers laquelle une instance peut être mise à niveau.

Type : chaîne

Obligatoire : non

Engine

Nom du moteur de base de données cible mis à niveau.

Type : chaîne

Obligatoire : non

EngineVersion

Numéro de version du moteur de base de données cible mis à niveau.

Type : chaîne

Obligatoire : non

IsMajorVersionUpgrade

Valeur qui indique si un moteur de base de données est mis à niveau vers une version majeure.

Type : booléen

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

VpcSecurityGroupMembership

Service : Amazon DocumentDB (with MongoDB compatibility)

Utilisé comme élément de réponse pour les requêtes concernant l'appartenance à un groupe de sécurité du cloud privé virtuel (VPC).

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

Status

Statut du groupe de sécurité VPC.

Type : chaîne

Obligatoire : non

VpcSecurityGroupId

Nom du groupe de sécurité VPC.

Type : chaîne

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Clusters Amazon DocumentDB Elastic

Les types de données suivants sont pris en charge par Amazon DocumentDB Elastic Clusters :

- Cluster
- ClusterInList
- ClusterSnapshot
- ClusterSnapshotInList
- Shard
- ValidationExceptionField

Cluster

Service : Amazon DocumentDB Elastic Clusters

Renvoie des informations sur un cluster élastique spécifique.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

adminUserName

Nom de l'administrateur du cluster Elastic.

Type : chaîne

Obligatoire : oui

authType

Type d'authentification pour le cluster élastique.

Type : chaîne

Valeurs valides : `PLAIN_TEXT | SECRET_ARN`

Obligatoire : oui

clusterArn

L'identifiant ARN du cluster élastique.

Type : chaîne

Obligatoire : oui

clusterEndpoint

URL utilisée pour se connecter au cluster élastique.

Type : chaîne

Obligatoire : oui

clusterName

Nom du cluster élastique.

Type : chaîne

Obligatoire : oui

createTime

Heure à laquelle le cluster élastique a été créé en temps universel coordonné (UTC).

Type : chaîne

Obligatoire : oui

kmsKeyId

Identifiant de clé KMS à utiliser pour chiffrer le cluster élastique.

Type : chaîne

Obligatoire : oui

preferredMaintenanceWindow

Intervalle de temps hebdomadaire, au format Universal Coordinated Time (UTC), pendant lequel a lieu la maintenance du système.

Format : `ddd:hh24:mi-ddd:hh24:mi`

Type : chaîne

Obligatoire : oui

shardCapacity

Le nombre de vCPU assignés à chaque partition de cluster élastique. Le maximum est de 64. Les valeurs autorisées sont 2, 4, 8, 16, 32, 64.

Type : entier

Obligatoire : oui

shardCount

Le nombre de partitions attribuées au cluster élastique. Le maximum est de 32.

Type : entier

Obligatoire : oui

status

État du cluster élastique.

Type : chaîne

Valeurs valides : `CREATING | ACTIVE | DELETING | UPDATING | VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED | INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID | INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN | INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING | SPLITTING | COPYING | STARTING | STOPPING | STOPPED`

Obligatoire : oui

subnetIds

Les identifiants de sous-réseau Amazon EC2 pour le cluster élastique.

Type : tableau de chaînes

Obligatoire : oui

vpcSecurityGroupIds

Liste des groupes de sécurité VPC EC2 associés à ce cluster élastique.

Type : tableau de chaînes

Obligatoire : oui

backupRetentionPeriod

Nombre de jours pendant lesquels les instantanés automatiques sont conservés.

Type : entier

Obligatoire : non

preferredBackupWindow

Plage de temps quotidienne pendant laquelle les sauvegardes automatisées sont créées si les sauvegardes automatisées sont activées, comme déterminé par`backupRetentionPeriod`.

Type : chaîne

Obligatoire : non

shardInstanceCount

Le nombre d'instances de répliques s'appliquant à toutes les partitions du cluster. Une `shardInstanceCount` valeur de 1 signifie qu'il existe une instance d'écriture et que toutes les instances supplémentaires sont des répliques qui peuvent être utilisées pour les lectures et pour améliorer la disponibilité.

Type : entier

Obligatoire : non

shards

Nombre total de partitions dans le cluster.

Type : tableau d’objets Shard

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

ClusterInList

Service : Amazon DocumentDB Elastic Clusters

Liste des clusters élastiques Amazon DocumentDB.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

clusterArn

L'identifiant ARN du cluster élastique.

Type : chaîne

Obligatoire : oui

clusterName

Nom de l'agrégat élastique.

Type : chaîne

Obligatoire : oui

status

État du cluster élastique.

Type : chaîne

Valeurs valides : `CREATING | ACTIVE | DELETING | UPDATING | VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED | INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID | INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN | INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING | SPLITTING | COPYING | STARTING | STOPPING | STOPPED`

Obligatoire : oui

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

ClusterSnapshot

Service : Amazon DocumentDB Elastic Clusters

Renvoie des informations sur un instantané de cluster élastique spécifique.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

adminUserName

Nom de l'administrateur du cluster Elastic.

Type : chaîne

Obligatoire : oui

clusterArn

L'identifiant ARN du cluster élastique.

Type : chaîne

Obligatoire : oui

clusterCreationTime

Heure à laquelle le cluster élastique a été créé en temps universel coordonné (UTC).

Type : chaîne

Obligatoire : oui

kmsKeyId

L'identifiant de clé KMS est l'Amazon Resource Name (ARN) de la clé de chiffrement KMS. Si vous créez un cluster en utilisant le même compte Amazon qui possède cette clé de chiffrement KMS, vous pouvez utiliser l'alias de clé KMS au lieu de l'ARN comme clé de chiffrement KMS. Si aucune clé de chiffrement n'est spécifiée ici, Amazon DocumentDB utilise la clé de chiffrement par défaut créée par KMS pour votre compte. Votre compte possède une clé de chiffrement par défaut différente pour chaque région Amazon.

Type : chaîne

Obligatoire : oui

snapshotArn

Identifiant ARN de l'instantané du cluster élastique.

Type : chaîne

Obligatoire : oui

snapshotCreationTime

Heure à laquelle l'instantané du cluster élastique a été créé en temps universel coordonné (UTC).

Type : chaîne

Obligatoire : oui

snapshotName

Nom de l'instantané du cluster élastique.

Type : chaîne

Obligatoire : oui

status

État de l'instantané du cluster élastique.

Type : chaîne

Valeurs valides : `CREATING | ACTIVE | DELETING | UPDATING | VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED | INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID | INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN | INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING | SPLITTING | COPYING | STARTING | STOPPING | STOPPED`

Obligatoire : oui

subnetIds

Les identifiants de sous-réseau Amazon EC2 pour le cluster élastique.

Type : tableau de chaînes

Obligatoire : oui

vpcSecurityGroupIds

Liste des groupes de sécurité VPC EC2 à associer au cluster élastique.

Type : tableau de chaînes

Obligatoire : oui

snapshotType

Type de snapshots de cluster à renvoyer. Vous pouvez spécifier l'une des valeurs suivantes :

- `automated`- Renvoie tous les instantanés de cluster qu'Amazon DocumentDB a automatiquement créés pour AWS votre compte.
- `manual`- Renvoie tous les instantanés de cluster que vous avez créés manuellement pour votre AWS compte.

Type : chaîne

Valeurs valides : `MANUAL | AUTOMATED`

Obligatoire : non

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

ClusterSnapshotInList

Service : Amazon DocumentDB Elastic Clusters

Liste des instantanés d'Elastic Cluster.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

clusterArn

L'identifiant ARN du cluster élastique.

Type : chaîne

Obligatoire : oui

snapshotArn

Identifiant ARN de l'instantané du cluster élastique.

Type : chaîne

Obligatoire : oui

snapshotCreationTime

Heure à laquelle l'instantané du cluster élastique a été créé en temps universel coordonné (UTC).

Type : chaîne

Obligatoire : oui

snapshotName

Nom de l'instantané du cluster élastique.

Type : chaîne

Obligatoire : oui

status

État de l'instantané du cluster élastique.

Type : chaîne

Valeurs valides : `CREATING | ACTIVE | DELETING | UPDATING | VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED | INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID | INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN | INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING | SPLITTING | COPYING | STARTING | STOPPING | STOPPED`

Obligatoire : oui

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Shard

Service : Amazon DocumentDB Elastic Clusters

Le nom du shard.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

createTime

Heure à laquelle la partition a été créée en temps universel coordonné (UTC).

Type : chaîne

Obligatoire : oui

shardId

L'ID du shard.

Type : chaîne

Obligatoire : oui

status

État actuel de la partition.

Type : chaîne

Valeurs valides : `CREATING | ACTIVE | DELETING | UPDATING | VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED | INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID | INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN | INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING | SPLITTING | COPYING | STARTING | STOPPING | STOPPED`

Obligatoire : oui

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

ValidationExceptionField

Service : Amazon DocumentDB Elastic Clusters

Champ spécifique dans lequel une exception de validation donnée s'est produite.

Table des matières

> **Note**
>
> Dans la liste suivante, les paramètres requis sont décrits en premier.

message

Un message d'erreur décrivant l'exception de validation dans ce champ.

Type : chaîne

Obligatoire : oui

name

Nom du champ dans lequel l'exception de validation s'est produite.

Type : chaîne

Obligatoire : oui

consultez aussi

Pour plus d'informations sur l'utilisation de cette API dans l'un des AWS SDK spécifiques au langage, consultez les pages suivantes :

- AWS SDK pour C++
- AWS SDK pour Java V2
- AWS SDK pour Ruby V3

Erreurs courantes

Cette section répertorie les erreurs communes aux actions d'API de tous les services AWS. Pour les erreurs spécifiques à une action d'API pour ce service, consultez la rubrique pour cette action d'API.

AccessDeniedException

Vous ne disposez pas d'un accès suffisant pour effectuer cette action.

Code d'état HTTP : 400

IncompleteSignature

La signature de la requête n'est pas conforme aux normes AWS.

Code d'état HTTP : 400

InternalFailure

Le traitement de la demande a échoué en raison d'une erreur, d'une exception ou d'un échec inconnu.

Code d'état HTTP : 500

InvalidAction

L'action ou l'opération demandée n'est pas valide. Vérifiez que l'action est entrée correctement.

Code d'état HTTP : 400

InvalidClientTokenId

Le certificat X.509 ou l'ID de clé d'accès AWS fourni(e) n'existe pas dans nos archives.

Code d'état HTTP : 403

NotAuthorized

Vous ne disposez pas de l'autorisation nécessaire pour effectuer cette action.

Code d'état HTTP : 400

OptInRequired

L’ID de clé d’accès AWS a besoin d’un abonnement pour le service.

Code d'état HTTP : 403

RequestExpired

La demande a atteint le service plus de 15 minutes après la date affichée sur la demande ou plus de 15 minutes après la date d'expiration de la demande (comme pour les URL pré-signées) ou la date affichée sur la demande est postérieure de 15 minutes.

Code d'état HTTP : 400

ServiceUnavailable

La requête a échoué en raison d'une défaillance temporaire du serveur.

HTTP Status Code: 503

ThrottlingException

La demande a été refusée suite à une limitation des demandes.

Code d'état HTTP : 400

ValidationError

L'entrée ne satisfait pas les contraintes spécifiées par un service AWS.

Code d'état HTTP : 400

Paramètres communs

La liste suivante contient les paramètres que toutes les actions utilisent pour signer les demandes Signature Version 4 à l'aide d'une chaîne de requête. Tous les paramètres spécifiques d'une action particulière sont énumérées dans le sujet consacré à cette action. Pour plus d'informations sur Signature version 4, consultez la section Signature de demandes d'AWSAPI dans le Guide de l'utilisateur IAM.

Action

Action à effectuer.

Type : chaîne

Obligatoire : oui

Version

Version de l'API pour laquelle la demande est écrite, au format AAAA-MM-JJ.

Type : chaîne

Obligatoire : oui

X-Amz-Algorithm

Algorithme de hachage que vous avez utilisé pour créer la signature de la demande.

Condition : spécifiez ce paramètre lorsque vous incluez des informations d'authentification dans une chaîne de requête plutôt que dans l'en-tête d'autorisation HTTP.

Type : chaîne

Valeurs valides : `AWS4-HMAC-SHA256`

Obligatoire : Conditionnelle

X-Amz-Credential

Valeur de la portée des informations d'identification, qui est une chaîne incluant votre clé d'accès, la date, la région cible, le service demandé et une chaîne de terminaison (« aws4_request »). Spécifiez la valeur au format suivant : access_key/AAAAMMJJ/région/service/aws4_request.

Pour plus d'informations, consultez la section Création d'une demande d'AWSAPI signée dans le Guide de l'utilisateur IAM.

Condition : spécifiez ce paramètre lorsque vous incluez des informations d'authentification dans une chaîne de requête plutôt que dans l'en-tête d'autorisation HTTP.

Type : chaîne

Obligatoire : Conditionnelle

X-Amz-Date

La date utilisée pour créer la signature. Le format doit être au format de base ISO 8601 (AAAAMMJJ'T'HHMMSS'Z'). Par exemple, la date/heure suivante est une valeur X-Amz-Date valide : `20120325T120000Z`.

Condition : X-Amz-Date est un en-tête facultatif pour toutes les demandes. Il peut être utilisé pour remplacer la date dans la signature des demandes. Si l'en-tête Date est spécifié au format de base ISO 8601, X-Amz-Date n'est pas obligatoire. Lorsque X-Amz-Date est utilisé, il remplace toujours la valeur de l'en-tête Date. Pour plus d'informations, consultez la section Éléments d'une signature de demande d'AWSAPI dans le Guide de l'utilisateur IAM.

Type : chaîne

Obligatoire : Conditionnelle

X-Amz-Security-Token

Le jeton de sécurité temporaire obtenu lors d'un appel àAWS Security Token Service (AWS STS). Pour obtenir la liste des services prenant en charge les informations d'identification deAWS STS sécurité temporaires issues de IAM dans le Guide de l'utilisateur IAM.Services AWS

Condition : si vous utilisez des informations d'identification de sécurité temporaires issues deAWS STS, vous devez inclure le jeton de sécurité.

Type : chaîne

Obligatoire : Conditionnelle

X-Amz-Signature

Spécifie la signature codée en hexadécimal qui a été calculée à partir de la chaîne à signer et de la clé de signature dérivée.

Condition : spécifiez ce paramètre lorsque vous incluez des informations d'authentification dans une chaîne de requête plutôt que dans l'en-tête d'autorisation HTTP.

Type : chaîne

Obligatoire : Conditionnelle

X-Amz-SignedHeaders

Spécifie tous les en-têtes HTTP qui ont été inclus dans la demande canonique. Pour plus d'informations sur la spécification d'en-têtes signés, consultez la section Création d'une demande d'AWSAPI signée dans le Guide de l'utilisateur IAM.

Condition : spécifiez ce paramètre lorsque vous incluez des informations d'authentification dans une chaîne de requête plutôt que dans l'en-tête d'autorisation HTTP.

Type : chaîne

Obligatoire : Conditionnelle

Notes de mise à jour

Ces notes de publication décrivent les fonctionnalités, les améliorations et les corrections de bogues d'Amazon DocumentDB par date de sortie. Les notes de publication incluent les mises à jour de toutes les versions du moteur Amazon DocumentDB au fur et à mesure de leur publication.

Vous pouvez déterminer la version actuelle du correctif du moteur Amazon DocumentDB en exécutant la commande suivante :

```
db.runCommand({getEngineVersion: 1})
```

Si votre cluster n'utilise pas la dernière version du moteur, il est probable que vous disposiez d'une maintenance en attente qui permettra de mettre à niveau votre moteur. Pour plus d'informations, consultez Gestion d'Amazon DocumentDB le guide du développeur.

Rubriques

- 3 août 2023
- 13 juillet 2023
- 7 juin 2023
- 10 mai 2023
- 4 avril 2023
- 22 mars 2023
- 1er mars 2023
- 27 février 2023
- 2 février 2023
- 30 novembre 2022
- 9 août 2022
- 25 juillet 2022
- 27 juin 2022
- 29 avril 2022
- 7 avril 2022
- 16 mars 2022
- 8 février 2022
- 24 janvier 2022
- 21 janvier 2022
- 25 octobre 2021
- 24 juin 2021
- 4 mai 2021
- 15 janvier 2021
- 9 novembre 2020
- 30 octobre 2020
- 22 septembre 2020
- 10 juillet 2020
- 30 juin 2020

29 mai 2024

> **Note**
>
> Le correctif du moteur Amazon DocumentDB suivant est en cours de distribution dans toutes les régions Amazon DocumentDB au cours des prochaines semaines. Lorsque ce correctif moteur sera disponible dans votre région, vous recevrez une notification de correctif de service via le AWS Health Dashboard (AHD) AWS Management Console et par e-mail à l'adresse e-mail de l'utilisateur root de votre AWS compte.
> Ce correctif du moteur inclut les nouvelles fonctionnalités et corrections de bogues suivantes. Veuillez noter que la liste ci-dessous, ainsi que la documentation de support pertinente, peuvent être mises à jour pour inclure des annonces de fonctionnalités supplémentaires une fois que le correctif moteur sera disponible dans toutes les régions.

Nouvelles fonctionnalités

Amazon DocumentDB 5.0 (correctif moteur version 3.0.6742)

- Support supplémentaire pour les `regexFind` opérateurs `regexMatch` et les opérateurs.
- Support supplémentaire pour garantir une précision totale dans les journaux d'audit lors du traitement de grands nombres entiers. Les journaux d'audit conservent désormais la représentation numérique exacte de tous les nombres, évitant ainsi toute perte de précision.

Amazon DocumentDB 4.0 (correctif moteur version 2.0.10593)

- Support supplémentaire pour garantir une précision totale dans les journaux d'audit lors du traitement de grands nombres entiers. Les journaux d'audit conservent désormais la représentation numérique exacte de tous les nombres, évitant ainsi toute perte de précision.

3 avril 2024

Amazon DocumentDB est désormais disponible dans la région Moyen-Orient (Émirats arabes unis). Pour plus d'informations, consultez ce billet de blog.

Nouvelles fonctionnalités

Amazon DocumentDB 5.0 (correctif moteur version 3.0.5721)

- Support ajouté `bypassDocumentValidation` et message d'erreur détaillé pour`$jsonSchema`. Pour plus d'informations sur `bypassDocumentValidation`, consultez bypassDocumentValidation.
- Ajout du support de`$expr`.
- Ajout de la prise en charge des jointures non corrélées. `$lookup`
- Ajout du support pour conserver les règles de validation lors de la phase `$out` d'agrégation.

Amazon DocumentDB 4.0 (correctif moteur version 2.0.10392)

- Ajout du support `bypassDocumentValidation` pour`$jsonSchema`. Pour plus d'informations sur `bypassDocumentValidation`, consultez bypassDocumentValidation.
- Ajout du support de`$expr`.
- Ajout de la prise en charge des jointures non corrélées. `$lookup`
- Ajout du support pour conserver les règles de validation lors de la phase `$out` d'agrégation.

Corrections de bogues et autres modifications

- Correction d'une erreur lors de l'invocation `db.coll.stats()` sur le shell mongo 1.7 et versions ultérieures.
- Correction d'un problème de fuite de mémoire pour les requêtes de flux de modifications faisant partie du même pipeline d'agrégation. `$regex`

22 février 2024

Nouvelles fonctionnalités

Clusters élastiques Amazon DocumentDB

Les clusters élastiques Amazon DocumentDB prennent désormais en charge les fonctionnalités suivantes :

- Répliques d'instances de partitions secondaires lisibles : pour plus d'informations, reportez-vous à l'étape 5b de. Étape 1 : Création d'un cluster élastique
- Démarrer/arrêter le cluster : pour plus d'informations, voir. Arrêt et démarrage d'un cluster élastique Amazon DocumentDB
- Instances de partition configurables : pour plus d'informations, reportez-vous à l'étape 5b de. Étape 1 : Création d'un cluster élastique
- Sauvegardes automatiques pour les instantanés : pour plus d'informations, voirGestion d'une sauvegarde automatique des instantanés d'un cluster Elastic.
- Copier un instantané : pour plus d'informations, voirCopier un instantané d'un cluster élastique.

30 janvier 2024

Nouvelles fonctionnalités

Clusters élastiques Amazon DocumentDB

Les clusters élastiques Amazon DocumentDB sont désormais disponibles dans les régions suivantes :

- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Séoul)
- Amérique du Sud (São Paulo)
- Europe (Londres)

Pour plus d’informations, consultez Région du cluster élastique et disponibilité des versions.

Clusters globaux Amazon DocumentDB

Les clusters mondiaux sont désormais disponibles dans les deux AWS GovCloud (US) régions : AWS GovCloud (USA Est) et AWS GovCloud (USA Ouest).

10 janvier 2024

Nouvelles fonctionnalités

Amazon DocumentDB 5.0 (versions du correctif moteur 3.0.4574, 3.0.4780, 3.0.4960)

- Ajout du support pour les index vectoriels HNSW. Pour plus d'informations, consultez Recherche vectorielle pour Amazon DocumentDB.
- Ajout d'un support pour les index partiels. Pour plus d'informations, consultez Index partiel.
- Ajout d'un support pour l'exécution GC sur une collection sous `currentOp` commande.
- Ajout de la prise en charge de l'index de texte pour la recherche de texte natif sur Amazon DocumentDB. Pour plus d'informations, consultez Effectuer une recherche de texte avec Amazon DocumentDB.
- Ajout de la prise en charge des mots clés de `$jsonSchema` schéma `type` `allOf` `oneOf` `anyOf` `notmaxItems`,`minItems`,`maxProperties`,`minProperties`,`pattern`,,`patternProperties`,`multip etuniqueItems`.

  Pour plus d'informations, consultez Utilisation de la validation du schéma JSON.
- Ajout de la prise en charge des opérateurs arithmétiques `$ceil` `$floor$ln`,,`$log`,`$log10`,`$sqrt`, et`$exp`.

  Pour plus d'informations, consultez Opérateurs arithmétiques.
- Ajout de la prise en charge de l'opérateur d'expression conditionnelle`$switch`.
- Ajout du support pour les constructions d'index `IVFFLAT` vectoriels parallèles. La documentation a été mise à jour en supprimant la limitation des builds d'index `IVFFLAT` vectoriels parallèles du guide du développeur.

Amazon DocumentDB 4.0 (versions du correctif moteur 2.0.10124, 2.0.10179, 2.0.10221)

- Ajout d'un support pour l'exécution GC sur une collection sous `currentOp` commande.
- Ajout de la prise en charge des mots clés de `$jsonSchema` schéma `type` `allOf` `oneOf` `anyOf` `notmaxItems`,`minItems`,`maxProperties`,`minProperties`,`pattern`,,`patternProperties`,`multip etuniqueItems`.

  Pour plus d'informations, consultez Utilisation de la validation du schéma JSON.
- Ajout de la prise en charge des opérateurs arithmétiques `$ceil` `$floor$ln`,,`$log`,`$log10`,`$sqrt`, et`$exp`.

  Pour plus d'informations, consultez Opérateurs arithmétiques.
- Ajout de la prise en charge de l'opérateur d'expression conditionnelle`$switch`.

Corrections de bogues et autres modifications

- Ajout d'une fonctionnalité d'invocation qui ne tient pas compte des majuscules et minuscules. `db.runCommand("dbstats")` Les clients Amazon DocumentDB 5.0 et 4.0 utilisant des versions de correctifs de moteur antérieures à 3.0.4960 ou 2.0.10221 doivent appliquer ces derniers correctifs de moteur.
- Correction d'une erreur lors de l'invocation `db.coll.stats()` sur le shell mongo 1.7 et versions ultérieures. La documentation a été mise à jour en supprimant le conseil de `db.coll.stats()` dépannage du shell mongo du guide du développeur.

20 décembre 2023

Autres modifications

Support activé pour la mise à niveau des versions majeures sur place dans Amazon DocumentDB 3.6 et 4.0. Pour plus d'informations, consultez Mise à niveau sur place de la version majeure d'Amazon DocumentDB.

13 décembre 2023

Nouvelles fonctionnalités

Ajout de la prise en charge de la connectivité EC2 en un clic. Pour plus d'informations, consultez Connectez-vous à l'aide d'Amazon EC2.

29 novembre 2023

Amazon DocumentDB 5.0 (correctif moteur version 3.0.3727)

Nouvelles fonctionnalités

Ajout du support pour la recherche vectorielle. Pour plus d'informations, consultez ce billet de blog et consultez le manuel Recherche vectorielle pour Amazon DocumentDB Amazon DocumentDB Developer Guide.

21 novembre 2023

Amazon DocumentDB 5.0 (correctif moteur version 3.0.3727)

Nouvelles fonctionnalités

Ajout de la prise en charge du stockage optimisé pour les E/S. Pour plus d'informations, consultez Configurations de stockage en cluster Amazon DocumentDB le manuel Amazon DocumentDB Developer Guide.

Intégration ajoutée pour l'apprentissage automatique sans code avec SageMaker Canvas. Pour plus d'informations, consultez Apprentissage automatique sans code avec Amazon Canvas SageMaker le manuel Amazon DocumentDB Developer Guide.

17 novembre 2023

Nouvelles fonctionnalités

Amazon DocumentDB est désormais disponible dans la région AWS GovCloud (USA Est). Pour plus d'informations, consultez ce billet de blog.

Corrections de bogues et autres modifications

Amazon DocumentDB 3.6 (version du correctif moteur 1.0.208570)

Les noms de variables locales définis par l'utilisateur prennent désormais en charge « _ » (trait de soulignement) pour les opérateurs de projection tels que `$let` et`$filter`.

6 novembre 2023

Amazon DocumentDB 5.0 (correctif moteur version 3.0.3727) et 4.0 (correctif moteur version 2.0.9876)

Nouvelles fonctionnalités

- Ajout de la prise en charge des mots clés de `$jsonSchema` schéma `maxLength` `minLength` `maximumminimum`,`exclusiveMaximum`,,`exclusiveMinimum`,`items`,, et`additionalItems`.

  Notez que la validation du schéma JSON n'est prise en charge que dans les clusters basés sur des instances.

- Ajout de la prise en charge de l'opérateur de pipeline d'`$convert`agrégation et de ses opérateurs dérivés `$toBool` abrégés`$toInt`,`$toLong`,`$toDouble`,,`$toString`, `$toDecimal`$toObjectId, et. `$toDate`
- Ajout de la prise en charge des opérateurs d'expression définis `$setDifference`$anyElementTrue, et`$allElementTrue`.

Corrections de bogues et autres modifications

Correction d'un problème qui `NaN` empêchait l'affichage d'une mise `-NaN` à jour du flux de modifications allant de à.

20 octobre 2023

Autres modifications

Amazon DocumentDB a identifié un problème et interdit temporairement les mises à niveau de versions majeures (MVU) dans toutes les régions. Nous avons identifié la cause première du problème et avons développé un correctif qui est actuellement en cours de test. Nous prévoyons que ce correctif sera déployé dans toutes les régions avant la fin du quatrième trimestre 2023. Le MVU restera désactivé jusqu'à ce que le correctif soit déployé dans toutes les régions. Consultez cette page de note de publication pour plus d'informations sur la disponibilité des fonctionnalités MVU.

En attendant, vous pouvez effectuer des mises AWS DMS à niveau de versions majeures en migrant votre base de données Amazon DocumentDB d'un cluster de version inférieure vers une version supérieure. Suivez les étapes décrites pour effectuer la mise Mise à niveau de votre cluster Amazon DocumentDB à l'aide de AWS Database Migration Service à niveau à l'aide de AWS DMS. Vous pouvez également consulter ce billet de blog pour plus d'informations sur les meilleures pratiques à suivre lors de la mise à niveau en utilisant AWS DMS.

25 septembre 2023

Nouvelles fonctionnalités

Amazon DocumentDB est désormais disponible dans la région Asie-Pacifique (Hong Kong). Pour plus d'informations, consultez ce billet de blog.

20 septembre 2023

Nouvelles fonctionnalités

Ajout de la prise en charge des mises à niveau des versions majeures sur place dans Amazon DocumentDB 3.6 et 4.0. Pour plus d'informations, consultez Mise à niveau sur place de la version majeure d'Amazon DocumentDB.

15 septembre 2023

Nouvelles fonctionnalités

Amazon DocumentDB 5.0 (correctif moteur version 3.0.3140) et 4.0 (correctif moteur version 2.0.9686)

- Ajout de la prise en charge du validateur de schéma $JSONSchema dans les clusters basés sur des instances uniquement.

  Pour plus d'informations, consultez Utilisation de la validation du schéma JSON.

11 septembre 2023

Nouvelles fonctionnalités

Amazon DocumentDB est désormais disponible dans la région Asie-Pacifique (Hyderabad). Pour plus d'informations, consultez ce billet de blog.

3 août 2023

Nouvelles fonctionnalités

Clusters élastiques Amazon DocumentDB

- Les clusters Amazon DocumentDB Elastic prennent désormais en charge les opérations suivantes :
  - `top`

- `collStats`
- `hint`
- `dataSize`

Consultez API MongoDB, opérations et types de données pris en charge la liste complète des commandes et opérations prises en charge.

- Les index Time to Live (TTL) sont désormais pris en charge.
- `hints`Les index sont désormais pris en charge par des expressions d'index.

13 juillet 2023

Nouvelles fonctionnalités

Amazon DocumentDB 5.0 (correctif moteur version 3.0.1948)

- Ajout de la prise en charge de la compression de documents.
- Ajout du support pour les builds d'index parallèles.
- Ajout de la prise en charge de l'état de construction de l'index.

Amazon DocumentDB 4.0 (correctif moteur version 2.0.9259)

- Ajout du support pour les builds d'index parallèles.

Corrections de bogues et autres modifications

Amazon DocumentDB 5.0 (correctif moteur version 3.0.1948)

- Correction d'un problème d'authentification lié `createCollection` aux clusters élastiques Amazon DocumentDB lorsque les utilisateurs n'ont pas accès aux collections du système.
- Problème résolu : les instances de région secondaire ne pouvaient pas utiliser les mêmes noms d'instance de région principale.

Amazon DocumentDB 4.0 (correctif moteur version 2.0.9259)

- L'ajout de requêtes de surveillance interne aux journaux d'audit a été arrêté.

7 juin 2023

Corrections de bogues et autres modifications

Amazon DocumentDB 5.0

- Les instances r5 et t3.medium sont désormais prises en charge dans Amazon DocumentDB 5.0.
- `engineVersion`l'option par défaut est `5.0.0` dans le AWS SDK AWS CLI, et AWS CloudFormation.

10 mai 2023

Corrections de bogues et autres modifications

Amazon DocumentDB 5.0 (correctif moteur version 3.0.1361)

- Ajout de la prise `ignoreunknownindexoptions` en charge de la createIndex commande.
- L'ajout de requêtes de surveillance interne aux journaux d'audit a été arrêté.
- Les noms de variables locales définis par l'utilisateur prennent désormais en charge « _ » (trait de soulignement) pour les opérateurs de projection tels que `$let` et`$filter`.

4 avril 2023

Corrections de bogues et autres modifications

Amazon DocumentDB 4.0 (correctif moteur version 2.0.8934)

- Correction d'un problème lié à l'audit DML lorsqu'il est activé pendant une charge de travail continue.
- Correction d'un problème lié à l'audit DML lorsque des commandes d'agrégation comportant un indice reçoivent une valeur de chaîne.
- Correction d'un problème de non-fonctionnement de `listCollections` la commande lorsque les utilisateurs ayant le rôle readwriteanydatabase avaient à la fois les options AuthorizedCollections et NameOnly définies sur true.
- Correction d'un problème d'analyse correcte de la chaîne numérique dans le nom d'un champ.

- Annulez les curseurs de longue durée lorsqu'ils ont un impact sur la collecte des déchets.
- Les noms de variables locales définis par l'utilisateur prennent désormais en charge « _ » (trait de soulignement) pour les opérateurs de projection tels que `$let` et`$filter`.

22 mars 2023

Nouvelles fonctionnalités

Les clusters élastiques Amazon DocumentDB sont désormais disponibles dans les régions Asie-Pacifique (Singapour), Asie-Pacifique (Sydney) et Asie-Pacifique (Tokyo). Pour plus d'informations, consultez Région du cluster élastique et disponibilité des versions.

1er mars 2023

Nouvelles fonctionnalités

Amazon DocumentDB 5.0 (correctif moteur version 3.0.775)

- Présentation d'Amazon DocumentDB 5.0
  - Compatibilité avec MongoDB 5.0 (prise en charge des pilotes d'API MongoDB 5.0)
  - Support du chiffrement au niveau du champ (FLE) côté client. Vous pouvez désormais chiffrer les champs côté client avant d'écrire les données dans le cluster Amazon DocumentDB. Pour plus d'informations, voir Chiffrement au niveau des champs côté client
  - Nouveaux opérateurs d'agrégation :`$dateAdd`, `$dateSubtract`
- Limite de stockage augmentée à 128 TiB pour tous les clusters Amazon DocumentDB basés sur des instances et les clusters élastiques basés sur des partitions.
- Amazon DocumentDB 5.0 prend désormais en charge le scan d'index avec l'`$elemMatch`opérateur au premier niveau d'imbrication. Les analyses d'index sont prises en charge lorsque la requête ne comporte qu'un seul niveau de `$elemMatch` filtre et que la `$elemMatch` requête imbriquée ne prend pas en charge l'analyse d'index.

  Forme de requête qui prend en charge le scan d'index :

  ```
  db.foo.find( { "a": {$elemMatch: { "b": "xyz", "c": "abc"} } })
  ```

  Forme de requête qui ne prend pas en charge le scan d'index :

```
db.foo.find( { "a": {$elemMatch: { "b": {$elemMatch: { "d": "xyz", "e":
 "abc"} }} } })
```

27 février 2023

Corrections de bogues et autres modifications

Amazon DocumentDB 4.0

Ajout du support pour AWS Lambda. Pour plus d'informations, consultez la section Utilisation AWS Lambda avec Change Streams.

2 février 2023

Corrections de bogues et autres modifications

Amazon DocumentDB 3.6 (version du correctif moteur 1.0.208432)

- Correction d'un problème lié à l'audit DML lorsqu'il est activé pendant une charge de travail continue.
- Correction d'un problème lié à l'audit DML lorsque des commandes d'agrégation comportant un indice reçoivent une valeur de chaîne.
- Correction d'un problème de non-fonctionnement de `listCollections` la commande lorsque les utilisateurs ayant le rôle readwriteanydatabase avaient à la fois les options AuthorizedCollections et NameOnly définies sur true.
- Correction d'un problème d'analyse correcte de la chaîne numérique dans le nom d'un champ.
- Annulez les curseurs de longue durée lorsqu'ils ont un impact sur la collecte des déchets.

30 novembre 2022

Nouvelles fonctionnalités

Clusters élastiques Amazon DocumentDB

Les clusters élastiques Amazon DocumentDB sont un nouveau type de cluster Amazon DocumentDB qui permet aux utilisateurs de tirer parti des API de partitionnement MongoDB pour étendre leur

cluster. Les clusters élastiques gèrent pratiquement n'importe quel nombre de lectures et d'écritures avec une capacité de stockage de plusieurs pétaoctets en répartissant les données et le calcul entre plusieurs instances et volumes de calcul sous-jacents. Pour en savoir plus, consultez la section Utilisation des clusters élastiques Amazon DocumentDB.

9 août 2022

Nouvelles fonctionnalités

Amazon DocumentDB 3.6 (version du correctif moteur 1.0.208152) et 4.0

- Ajout du support pour le type de données Decimal128. Le Decimal128 est un type de données BSON pris en charge dans toutes les régions où DocumentDB est disponible.

  Pour plus d'informations, consultez Types de données.
- Ajout de la prise en charge de l'audit des requêtes DML avec Amazon CloudWatch Logs. Amazon DocumentDB peut désormais enregistrer des événements DML (Data Manipulation Language) et DDL (Data Definition Language) sur Amazon Logs. CloudWatch

  Pour plus d'informations, consultez ce billet de blog.

Corrections de bogues et autres modifications

Amazon DocumentDB 3.6 (version du correctif moteur 1.0.208152) et 4.0

- Vous pouvez désormais modifier votre propre mot de passe par votre propre mot de passe avec privilège. `changeOwnPassword`

25 juillet 2022

Nouvelles fonctionnalités

Amazon DocumentDB 4.0

Vous pouvez désormais créer des clusters plus rapidement grâce à la possibilité de créer des clones utilisant le même volume de cluster DocumentDB et contenant les mêmes données que le cluster d'origine. Pour plus de détails, consultez la section Gestion des clusters Amazon DocumentDB.

27 juin 2022

Nouvelles fonctionnalités

Amazon DocumentDB 4.0 (correctif moteur version 2.0.7509)

Amazon DocumentDB redimensionne dynamiquement votre base de données en fonction des modèles d'utilisation. L'ajout de données augmente l'espace jusqu'à 64 Tebioctets (TiB) et la suppression de données réduit l'espace alloué.

29 avril 2022

Nouvelles fonctionnalités

Amazon DocumentDB est désormais disponible dans la région Chine (Pékin). Pour plus d'informations, consultez ce billet de blog.

7 avril 2022

Nouvelles fonctionnalités

Amazon DocumentDB 3.6 (versions du correctif moteur 1.0.207836 et 1.0.208015) et 4.0 (versions du correctif moteur 2.0.6142 et 2.0.6948)

Amazon DocumentDB Performance Insights est désormais disponible en version préliminaire. Vous pouvez désormais enregistrer l'historique des performances sur sept jours dans une fenêtre mobile sans frais supplémentaires. Pour plus d'informations, consultez la section Surveillance avec Performance Insights.

16 mars 2022

Nouvelles fonctionnalités

Amazon DocumentDB est désormais disponible dans la région Europe (Milan). Pour plus d'informations, consultez ce billet de blog.

8 février 2022

Nouvelles fonctionnalités

Les instances Amazon DocumentDB R6g et T4g sont désormais disponibles en Asie-Pacifique, en Amérique du Sud et en Europe. Pour plus d'informations, consultez ce billet de blog.

24 janvier 2022

Nouvelles fonctionnalités

Amazon DocumentDB 3.6 (correctif moteur version 1.0.207684) et 4.0 (correctif moteur version 2.0.5170)

- DocDB ; propose désormais un essai gratuit. Pour plus de détails, consultez la page d'essai gratuit d'Amazon DocumentDB.
- Vous pouvez désormais utiliser des fonctionnalités améliorées avec les requêtes géospatiales, notamment les API suivantes :
  - `$geoWithin`
  - `$geoIntersects`
- Ajout du support pour les opérateurs MongoDB suivants :
  - `$mergeObjects`
  - `$reduce`

  Pour plus d'informations, consultez la section Interrogation de données géospatiales avec Amazon DocumentDB.

21 janvier 2022

Nouvelles fonctionnalités

Amazon DocumentDB 4.0 (correctif moteur version 2.0.5706)

- Les instances Amazon DocumentDB Graviton2 (r6g.large, r6g.2xlarge, r6g.4xlarge, r6g.8xlarge, r6g.12xlarge, r6g.16xlarge et t4g.medium) sont désormais prises en charge

Amazon DocumentDB 3.6 (correctif moteur version 1.0.207781) et 4.0 (correctif moteur version 2.0.5706)

- Ajout du support pour les API MongoDB suivantes :
  - `$reduce`
  - `$mergeObjects`
  - `$geoWithin`
  - `$geoIntersects`

25 octobre 2021

Nouvelles fonctionnalités

Amazon DocumentDB 3.6 (correctif moteur version 1.0.207780) et 4.0 (correctif moteur version 2.0.5704)

- Ajout du support pour les API MongoDB suivantes
  - `$literal`
  - `$map`
  - `$$ROOT`
- Support des fonctionnalités de GeoSpatial requête. Consultez ce billet de blog pour plus de détails
- Support pour le contrôle d'accès avec des rôles définis par l'utilisateur. Consultez ce billet de blog pour plus de détails
- Pilote Amazon DocumentDB JDBC pour permettre la connectivité à partir d'outils de BI tels que Tableau et d'outils de requête tels que SQL Workbench

Corrections de bogues et autres modifications

Amazon DocumentDB 3.6 (correctif moteur version 1.0.207780) et 4.0 (correctif moteur version 2.0.5704)

- Correction d'un bug permettant `$natural` de trier correctement lorsqu'un explicite `.sort()` est présent avec `$natural`
- Correction d'un bogue pour que le flux de modifications fonctionne avec `$redact`

- Correction d'un bogue `$ifNull` pour fonctionner avec un tableau vide
- Correction d'un bogue en cas de consommation excessive de ressources/de plantage du serveur lorsqu'un utilisateur actuellement connecté est supprimé ou que le privilège de cet utilisateur pour une activité en cours est révoqué
- Correction d'un bogue `listDatabase` et vérification des `listCollection` privilèges
- Correction d'un bug : logique de déduplication pour les éléments multiclés

24 juin 2021

Nouvelles fonctionnalités

Amazon DocumentDB 3.6 (correctif moteur version 1.0.207117) et 4.0 (correctif moteur version 2.0.3371)

- Les instances r5.8xlarge et r5.16xlarge sont désormais prises en charge. Pour en savoir plus, consultez le billet de blog Amazon DocumentDB supporte désormais les instances r5.8xlarge et r5.16xlarge.
- Les clusters mondiaux sont désormais pris en charge pour assurer la reprise après sinistre en cas de panne régionale et permettre des lectures globales à faible latence en autorisant les lectures depuis le cluster Amazon DocumentDB le plus proche.

4 mai 2021

Nouvelles fonctionnalités

Découvrez toutes les nouvelles fonctionnalités dans cet article de blog.

Amazon DocumentDB 3.6 (correctif moteur version 1.0.207117) et 4.0 (correctif moteur version 2.0.3371)

- `renameCollection`
- `$zip`
- `$indexOfArray`
- `$reverseArray`

- `$natural`
- `$hint`support pour la mise à jour
- Analyse de l'index pour `distinct`

Corrections de bogues et autres modifications

Amazon DocumentDB 3.6 (correctif moteur version 1.0.207117) et 4.0 (correctif moteur version 2.0.3371)

- Réduction de l'utilisation de la mémoire pour les `$in` requêtes
- Correction d'une fuite de mémoire dans les index multiclés
- Correction du plan d'explication et de la sortie du profileur pour `$out`
- Ajout d'un délai d'attente pour les opérations depuis le système de surveillance interne afin d'améliorer la fiabilité
- Correction d'un défaut affectant les prédicats de requête transmis aux index multiclés

15 janvier 2021

Nouvelles fonctionnalités

Amazon DocumentDB 4.0 (correctif moteur version 2.0.722)

- Aucun

Amazon DocumentDB 3.6 (version du correctif moteur 1.0.206295)

- Possibilité d'utiliser un index avec l'étape `$lookup` d'agrégation
- `find()`les requêtes avec projections peuvent être servies dans la direction d'un index (requête couverte)
- Possibilité d'utilisation `hint()` avec `findAndModify`
- Optimisations des performances pour l'opérateur `$addToSet`
- Améliorations visant à réduire la taille globale des index
- Nouveaux opérateurs d'agrégation : `$ifNull``$replaceRoot`,`$setIsSubset`,`$setInstersection`,`$setUnion`, et `$setEquals`

- Les utilisateurs peuvent également terminer leurs propres curseurs sans avoir besoin du rôle `KillCursor`

9 novembre 2020

Nouvelles fonctionnalités

Découvrez toutes les nouvelles fonctionnalités dans cet article de blog.

Amazon DocumentDB 4.0 (correctif moteur version 2.0.722)

- Compatibilité avec MongoDB 4.0
- Transactions ACID
- Support des flux de `(db.watch())` modification au niveau de la base de données `cluster(client.watch()` ou `mongo.watch())`
- Possibilité de démarrer ou de reprendre un flux de modifications en utilisant `startAtOperationTime`
- Prolongez la période de conservation de votre flux de modifications à 7 jours (24 heures auparavant)
- AWS DMS cible pour Amazon DocumentDB 4.0
- CloudWatch métriques : `TransactionsOpenTransactionsOpenMax`,`TransactionsAborted`,`TransactionsStarted`, et `TransactionsCommitted`
- Nouveaux champs pour les transactions dans `currentOpServerStatus`, et`profiler`.
- Possibilité d'utiliser un index avec l'étape `$lookup` d'agrégation
- `find()`les requêtes avec projections peuvent être servies dans la direction d'un index (requête couverte)
- Possibilité d'utilisation `hint()` avec `findAndModify`
- Optimisations des performances pour l'opérateur `$addToSet`
- Améliorations visant à réduire la taille globale des index.
- Nouveaux opérateurs d'agrégation : `$ifNull$replaceRoot`,`$setIsSubset`,`$setInstersection`,`$setUnion`, et `$setEquals`
- Avec les `ListDatabase` commandes `ListCollection` et, vous pouvez désormais éventuellement utiliser les `authorizedDatabases` paramètres `authorizedCollections` et

pour permettre aux utilisateurs de répertorier les collections et les bases de données auxquelles ils sont autorisés à accéder sans avoir besoin des `listDatabase` rôles `listCollections` et, respectivement.

- Les utilisateurs peuvent également terminer leurs propres curseurs sans avoir besoin du rôle `KillCursor`
- La comparaison des types numériques de sous-documents est désormais cohérente avec la comparaison des types numériques de documents de premier niveau. Le comportement dans Amazon DocumentDB 4.0 est désormais compatible avec MongoDB.

Amazon DocumentDB 3.6 (version du correctif moteur 1.0.206295)

- Aucun

Corrections de bogues et autres modifications

Amazon DocumentDB 4.0 (correctif moteur version 2.0.722)

- `$setOnInsert`n'autorise plus les mises à jour lors de l'utilisation de l'opérateur $ positionnel. Le comportement dans Amazon DocumentDB 4.0 est désormais compatible avec MongoDB.
- Correction d'un problème avec `$createCollection` et set `autoIndexId`
- Projection pour les documents imbriqués
- Modification du paramètre par défaut pour la mémoire de travail afin de l'adapter à la taille de la mémoire de l'instance
- Amélioration de la collecte des déchets
- Recherche avec clé vide dans le chemin, différence de comportement avec mongo
- Correction d'`dateToString`un bug dans le comportement du fuseau horaire
- Fixe `$push` (agrégation) pour respecter l'ordre de tri
- Correction d'un bug lié à `$currentOp` l'agrégat
- Correction d'un problème lié à `readPreference` l'absence de secondaire
- Correction d'un problème lié à la validation de `$createIndex` la même base de données que celle dans laquelle la commande a été émise
- Correction d'un comportement incohérent en cas `minKey` d'échec `maxKey` de la recherche
- Correction d'un problème lié au `$size` fait que l'opérateur ne fonctionnait pas avec une matrice composite

- Correction d'un problème lié à la négation de `$in` with regex
- Correction d'un problème lié à l'exécution d'une `$distinct` commande contre une vue
- Correction d'un problème lié aux commandes d'agrégation et de recherche qui triaient différemment les champs manquants
- Corrigé `$eq` au fait que l'expression régulière ne vérifiait pas le type
- Correction d'`$currentDate`un bug dans le comportement de la position ordinale de l'horodatage
- Granularité fixe en millisecondes pour `$currentDate`

Amazon DocumentDB 3.6 (version du correctif moteur 1.0.206295)

- Aucun

30 octobre 2020

Nouvelles fonctionnalités

Découvrez toutes les nouvelles fonctionnalités dans cet article de blog.

Amazon DocumentDB 3.6 (version du correctif moteur 1.0.206295)

- Ajout de la possibilité d'ouvrir un curseur de flux de modifications au niveau du cluster (`client.watch()` ou de `mongo.watch()`) la base de données (`db.watch()`)
- Possibilité d'augmenter la période de rétention du flux de modifications à 7 jours (24 heures auparavant)

Corrections de bogues et autres modifications

Amazon DocumentDB 3.6 (version du correctif moteur 1.0.206295)

- Diverses améliorations générales des performances du boîtier
- Une amélioration ciblée de la sécurité
- Correction d'un problème lié au tri par saut dans le deuxième champ d'un index composé
- Activer l'index normal pour l'égalité sur un seul champ d'un index à clés multiples (non composé)
- Condition de course à l'authentification fixe
- Correction d'un problème qui provoquait un crash peu fréquent lors de la collecte des ordures

- Amélioration de la sécurité RBAC
- `databaseConnectionsMax`Métrique ajoutée
- Améliorations des performances pour certaines charges de travail sur les instances `r5.24xlarge`

22 septembre 2020

Nouvelles fonctionnalités

Découvrez toutes les nouvelles fonctionnalités dans cet article de blog.

Amazon DocumentDB 3.6 (version du correctif moteur 1.0.206295)

- `$out`phase d'agrégation
- Le nombre maximum de connexions et de curseur par instance a été multiplié par 10

Corrections de bogues et autres modifications

Amazon DocumentDB 3.6 (version du correctif moteur 1.0.206295)

- Aucun

10 juillet 2020

Nouvelles fonctionnalités

Découvrez toutes les nouvelles fonctionnalités dans cet article de blog.

Amazon DocumentDB 3.6 (version du correctif moteur 1.0.206295)

- Copie instantanée entre régions

Corrections de bogues et autres modifications

Amazon DocumentDB 3.6 (version du correctif moteur 1.0.206295)

- Aucun

30 juin 2020

Nouvelles fonctionnalités

Découvrez toutes les nouvelles fonctionnalités dans cet article de blog.

Amazon DocumentDB 3.6 (version du correctif moteur 1.0.206295)

- Instances moyennes T3

Corrections de bogues et autres modifications

Amazon DocumentDB 3.6 (version du correctif moteur 1.0.206295)

- Récupération de mémoire inactive pour les instances t3
- Améliorations d'authentification
- Performances d'authentification SASL améliorées
- Correction d'un `currentOp` problème lié au dépassement du nombre maximum d'opérations possibles
- Correction d'un`killOps` problème de mise à jour et de suppression groupées
- Améliorations `$sample` des performances avec `$match`
- Support fixe pour, `$$` dans le second cas, en phase de rédaction
- Correction de diverses causes de crash récurrentes
- Améliorations apportées au balayage TTL pour réduire iOS et la latence
- Utilisation optimisée de la mémoire pour `$unwind`
- Statistiques de collecte fixes, état de course avec indice de chute
- Condition de course fixe lors de la création simultanée de l'index
- Correction d'un crash peu fréquent dans l'index `hash_search`

Historique du document pour le guide du développeur Amazon DocumentDB

- Version de l'API : 2014-10-31
- Dernière mise à jour de la documentation : 2 juin 2023

Le tableau suivant décrit la documentation de cette version du manuel Amazon DocumentDB Developer Guide.

Modification	Description	Date
AWS mise à jour des politiques gérée - modification des politiques	Amazon DocumentDB met à jour les politiques d'accès complet pour les clusters élastiques.	21 février 2024
AWS mise à jour des politiques gérée - modification des politiques	Amazon DocumentDB met à jour les politiques de lecture seule et d'accès complet pour les clusters élastiques.	21 juin 2023
AWS mise à jour des politiques gérée - nouvelle politique	Amazon DocumentDB introduit une nouvelle politique de lecture seule pour les clusters élastiques.	8 juin 2023
AWS mise à jour des politiques gérée - nouvelle politique	Amazon DocumentDB introduit une nouvelle politique d'accès complet pour les clusters élastiques.	5 juin 2023
Compatibilité avec MongoDB 5.0	Amazon DocumentDB est désormais compatible avec la version 5.0 de MongoDB.	1er mars 2023

Mise à jour des politiques	Pour prendre en charge la fonctionnalité de cluster élastique Amazon AmazonDoc DocumentDB, la ConsoleFu llAccess politique de base de données est mise à jour et le AmazonDoc DB- ElasticSe rviceRolePolicy est introduit.	30 novembre 2022
Clusters élastiques	Ajout d'une nouvelle fonctionn alité Elastic Cluster prenant en charge le partitionnement (sharding) basé sur le hachage des données dans le système de stockage distribué d'Amazon DocumentDB.	30 novembre 2022
Clusters mondiaux	Ajout de documentation sur l'utilisation des clusters globaux.	2 juin 2021
Abonnements aux événement s	Ajout de la documentation d'abonnement aux événement s.	26 mars 2021
Améliorations de la version 3.6	Améliorations documentées apportées à la version 3.6 en ce qui concerne les contrôles d'accès basés sur les rôles, les opérateurs d'agrégation et les performances.	15 janvier 2021
Compatibilité avec MongoDB 4.0	Amazon DocumentDB est désormais compatible avec la version 4.0 de MongoDB.	9 novembre 2020

Guides de démarrage	Nouveaux guides de démarrage pour démarrer avec Amazon DocumentDB à l'aide d'Amazon EC2 AWS Cloud9, Robo3T ou Studio3T.	15 août 2020
Zones de disponibilité supplémentaires prises en charge	Amazon DocumentDB a ajouté la prise en charge d'une zone de disponibilité supplémentaire en Asie-Pacifique (Séoul) (ap-northeast-2).	14 juillet 2020
Ajout de la prise en charge de la copie d'instantanés entre les régions.	Amazon DocumentDB a ajouté la prise en charge de la copie des instantanés de cluster. Régions AWS Pour plus d'informations, voir Copier des instantanés d'une région à l'autre.	10 juillet 2020
Ajout du support pour la classe d'instance T3.	Ajout de la prise en charge des types d'instances T3 dans toutes les régions prenant en charge Amazon DocumentDB. Pour plus d'informations, consultez les sections Classes d'instances prises en charge par région et Spécifications des classes d'instances.	30 juin 2020
Ajout du support pour AWS GovCloud (US).	Amazon DocumentDB est désormais disponible dans la AWS GovCloud (US) région (us-gov-west-1).	29 juin 2020

16 nouvelles CloudWatch métriques ont été ajoutées.	Amazon DocumentDB a ajouté la prise en charge de 16 nouvelles métriques Amazon CloudWatch . Pour plus d'informations, consultez la section Surveillance d'Amazon DocumentDB avec. CloudWatch	23 Juin 2020
Ajout du support pour les caractères nuls et l'opérateur $regex.	Amazon DocumentDB a ajouté la prise en charge des caractères nuls dans les chaînes et la possibili té d'utiliser un index pour $regex. Pour connaître les API MongoDB et les fonctionn alités de pipeline d'agrégation prises en charge pour Amazon DocumentDB, consultez la section Différences fonctionn elles avec MongoDB.	22 juin 2020
Ajout de la prise en charge des capacités d'indexation multiclés améliorées.	Amazon DocumentDB a ajouté la prise en charge de fonctionnalités d'indexat ion multiclés améliorées, notamment l'indexation de tableaux de plus de 2 048 octets et la possibilité de créer un index multiclé composé avec plusieurs clés dans le même tableau. Pour de plus amples informations, veuillez consulter Différences fonctionnelles avec MongoDB.	23 avril 2020

Ajout de la prise en charge de la protection contre la suppression pour une pile Amazon DocumentDB. AWS CloudFormation	Amazon DocumentDB a ajouté la prise en charge de l'activation de la protection contre la suppression lors de la création d'une pile Amazon AWS CloudFormation DocumentDB.	20 avril 2020
Ajout de la prise en charge du contrôle d'accès basé sur les rôles.	Amazon DocumentDB a ajouté la prise en charge du contrôle d'accès basé sur les rôles à l'aide de rôles intégrés.	26 mars 2020
Ajout du support pour une zone de disponibilité supplémentaire au Canada (Central) (ca-central-1).	Amazon DocumentDB est désormais disponible dans la région du Canada (Centre) (ca-central-1) avec des instances de classe R5 et 3 zones de disponibilité.	26 mars 2020
Ajout du support pour deux API MongoDB supplémentaires.	Amazon DocumentDB a ajouté la prise en charge des API MongoDB et `$dateFromString` MongoDB. `executionStats`	23 mars 2020
Ajout du support pour cinq API MongoDB supplémentaires.	Amazon DocumentDB a ajouté la prise en charge des `$objectToArray` API`$arrayToObject` ,`$slice`,`$mod`, et MongoDB. `$range`	6 février 2020
Ajout du support pour le Canada (Central).	Amazon DocumentDB est désormais disponible dans la région du Canada (Centre) (ca-central-1) avec des instances de classe R5.	11 décembre 2019

Ajout du support pour ChangeStreamLogSize.	Amazon DocumentDB a ajouté la prise en charge `ChangeStreamLogSize` des métriques Cloudwatch.	22 novembre 2019
Support supplémentaire pour la région Europe (Paris)	Amazon DocumentDB est désormais disponible dans la région Europe (Paris) (eu-west-3) avec des instances de classe R5.	30 octobre 2019
Ajout du support pour la région Asie-Pacifique (Mumbai)	Amazon DocumentDB est désormais disponible dans la région Asie-Pacifique (Mumbai) (ap-south-1) avec des instances de classe R5.	17 octobre 2019
Ajout du support pour trois API MongoDB supplémentaires	Amazon DocumentDB a ajouté la prise en charge des API`$addFields` ,`$concatArrays` , et MongoDB. `$lookup`	16 octobre 2019
Ajout du support pour la région Asie-Pacifique (Singapour)	Amazon DocumentDB est désormais disponible dans la région Asie-Pacifique (Singapour) (ap-southeast-1) avec des instances de classe R5.	14 octobre 2019
Ajout d'un nouveau document pour la mise à jour des certificats TLS	Ajout d'instructions pour la mise à jour des certificats d'autorité de certification afin d'utiliser le nouveau certificat d'autorité de certification pour créer des connexions TLS.	2 octobre 2019

Ajout de la prise en charge des API pour les certificats	Amazon DocumentDB est un nouveau type de données de certificat pour les instances . Pour plus d'informations, consultez DBInstance.	1 octobre 2019
Support pour le profilage des requêtes	Amazon DocumentDB a ajouté la possibilité de profiler les opérations prises en charge sur les instances et les bases de données de votre cluster.	19 août 2019
Ajout d'un troisième AZ en Asie-Pacifique (Tokyo)	Amazon DocumentDB a ajouté une troisième zone de disponibilité (AZ) pour vos instances de calcul en Asie-Pacifique (Tokyo).	9 août 2019
Support pour des API Mongo supplémentaires	Ajout de la prise en charge de fonctionnalités de pipeline d'agrégation supplémentaires `$in$isoWeek`, notamment les opérateurs `$isoWeekY ear $isoDayOfWeek` ,,,, et `$dateToString` la phase d'`$addToSet` agrégatio n. Amazon DocumentDB a également ajouté la prise en charge de la `top()` commande pour les diagnosti cs au niveau de la collecte et la possibilité de modifier le `expireAfterSeconds` paramètre des index TTL à l'aide de cette commande. `collMod()`	31 juillet 2019

Support supplémentaire pour l'Europe (Londres)	Amazon DocumentDB est désormais disponible en Europe (Londres) (eu-west-2) avec des instances de classe R5.	18 juillet 2019
Exemples de code ajoutés	Ajout d'exemples de code dans R et Ruby pour la connexion par programmation à Amazon DocumentDB.	17 juillet 2019
Meilleure pratique ajoutée	Ajout d'une meilleure pratique pour vous aider à gérer les coûts liés à Amazon DocumentDB.	17 juillet 2019
Support pour l'arrêt et le démarrage d'un cluster	Amazon DocumentDB a ajouté la prise en charge de l'arrêt et du démarrage des clusters afin de gérer les coûts des environnements de développement et de test.	1 juillet 2019

Support pour la protection contre la suppression des clusters	Pour protéger vos clusters contre toute suppressi on accidentelle, Amazon DocumentDB a ajouté une protection contre la suppressi on. Pour plus d'informa tions, consultez les rubriques suivantes : Création d'un cluster Amazon DocumentD B, Modification d'un cluster Amazon DocumentDB, Suppression d'un cluster Amazon DocumentDB, `DeletionProtection` ainsi que dans la rubrique d'API DBCluster.	1 juillet 2019
Mise à jour des différences fonctionnelles	Ajout de transactions implicites aux différences fonctionnelles.	26 juin 2019
Ajout de différences fonctionn elles	Ajout d'une remarque concernant le stockage et la compression d'index dans Amazon DocumentDB.	13 juin 2019
Région supplémentaire prise en charge	Amazon DocumentDB est désormais disponible en Asie-Pacifique (Sydney) (ap-south east-2) avec des instances de classe R5.	5 juin 2019

Classe d'instance R5 prise en charge dans d'autres régions	Ajout de la prise en charge des classes d'instances R5 pour les 4 régions supplémentaires : USA Est (Ohio), USA Est (Virginie du Nord), USA Ouest (Oregon) et UE (Irlande). Avec cette modification, les instances R5 sont prises en charge dans toutes les régions prenant en charge Amazon DocumentDB.	17 mai 2019
Autres régions prises en charge	Ajout du support pour 2 régions supplémentaires, Asie-Pacifique (Tokyo) (ap-northeast-1) et Asie-Pacifique (Séoul) (ap-northeast-2) avec les classes d'instance R5. Pour plus d'informations, consultez Classes d'instance prises en charge par région et Spécifications de classe d'instance.	8 mai 2019
Ajout d'autres exemples de code de connexion	Ajout d'exemples de code en Java et C# pour la connexion à Amazon DocumentDB.	24 avril 2019

Support supplémentaire de l'API Mongo	Ajout de la prise en charge de sept opérateurs de chaîne de regroupement (`$indexOfBytes` , `$indexOfCP` , `$strLenBytes` , `$strLenCP` , `$toLower`, `$toUpper` et `$split`), de neuf opérateurs date-heures (`$dayOfYear` , `$dayOfMonth` , `$dayOfWeek` , `$year`, `$month`, `$hour`, `$minute`, `$second` et `$millisecond`) et du pipeline de regroupement `$sample`.	4 avril 2019
Exemples de code de connexion ajoutés	Ajout d'exemples de code en Python, Node.js, PHP et Go pour la connexion à Amazon DocumentDB.	21 mars 2019
Support pour la région de Francfort et les instances R5	Ajout du support pour la région Europe (Francfort) (eu-central-1) avec les classes d'instance R5. Pour plus d'informations, consultez Classes d’instance prises en charge par région et Spécifications de classe d’instance.	13 mars 2019

Assistance aux opérateurs de pipelines d'agrégation	Ajout de la prise en charge des nouveaux opérateurs de chaîne de regroupement (`$concat`, `$substr`, `$substrBytes` , `$substrCP` , `$strcasecmp`), d'un opérateur de regroupement de tableau (`$size`), d'un opérateur d'accumulation de groupe de regroupement (`$push`) et des étapes de regroupement (`$redact` et `$indexStats`). Nous avons également ajouté la prise en charge des opérateurs de tableau de position (`$[]` et `$[<identifier>]`) et `hint()`.	28 février 2019
Améliorations du moteur	Ajout de la documentation afin de déterminer les modifications de cluster en attente et de mettre à niveau la version du moteur de votre cluster.	15 février 2019
Événements d'audit	Ajout de la prise en charge de l'audit des événements de base de données avec Amazon CloudWatch Logs.	12 février 2019
Quick Start	Ajout d'une rubrique de démarrage rapide pour vous aider à démarrer facilement avec Amazon DocumentDB en utilisant. AWS CloudFormation	11 janvier 2019

Publication publique	Il s'agit de la première version publique d'Amazon DocumentDB (compatible avec MongoDB). Cette version comprend le Manuel du développeur et la Référence d'API de gestion des ressources intégrée.	9 janvier 2019

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.