aws

Guía para desarrolladores

Amazon MemoryDB para Redis

Amazon MemoryDB para Redis: Guía para desarrolladores

Table of Contents

¿Qué es MemoryDB para Redis?

MemoryDB para Redis es un servicio de base de datos en memoria duradero que ofrece un rendimiento ultrarrápido. Está diseñado específicamente para aplicaciones modernas con arquitecturas de microservicios.

MemoryDB es compatible con Redis, un popular almacén de datos de código abierto, que le permite crear aplicaciones rápidamente utilizando las mismas estructuras de datos, API y comandos de Redis flexibles y fáciles de usar que ya se utilizan en la actualidad. Con MemoryDB, todos sus datos se almacenan en la memoria, lo que le permite lograr una latencia de lectura de microsegundos y una latencia de escritura de milisegundos de un solo dígito y un alto rendimiento. MemoryDB también almacena los datos de forma duradera en varias zonas de disponibilidad (AZ) mediante un registro transaccional multizona para permitir una rápida conmutación por error, la recuperación de la base de datos y el reinicio de los nodos.

MemoryDB, que ofrece un rendimiento en memoria y una durabilidad en zonas de disponibilidad múltiples, se puede utilizar como base de datos principal de alto rendimiento para sus aplicaciones de microservicios, lo que elimina la necesidad de gestionar por separado tanto la caché como la base de datos duradera.

Temas

Características de MemoryDB

MemoryDB para Redis es un servicio de base de datos en memoria duradero que ofrece un rendimiento ultrarrápido. Las características de MemoryDB incluyen:

- Consistencia sólida para los nodos principales y consistencia final garantizada para los nodos de réplica. Para obtener más información, consulte Coherencia.

- Latencias de lectura de microsegundos y de escritura de milisegundos de un solo dígito con hasta 160 millones de TPS por clúster.
- APIs y estructuras de datos de Redis flexibles y fáciles de usar. Cree nuevas aplicaciones o migre fácilmente las aplicaciones de Redis existentes prácticamente sin necesidad de modificarlas.
- Durabilidad de los datos mediante un registro transaccional Multi-AZ que proporciona una recuperación y un reinicio rápidos de la base de datos.
- Disponibilidad en zonas de disponibilidad múltiples (Multi-AZ) con conmutación por error automática y detección y recuperación de los fallos de los nodos.
- Escale fácilmente horizontalmente añadiendo y eliminando nodos o verticalmente desplazándose a tipos de nodos más grandes o más pequeños. Puede escalar el rendimiento de escritura añadiendo particiones y escalar el rendimiento de lectura añadiendo réplicas.
- Coherencia de lectura tras escritura para los nodos principales y coherencia final garantizada para los nodos de réplica.
- MemoryDB admite el cifrado en tránsito, el cifrado en reposo y la autenticación de usuarios mediante Autenticación de usuarios con listas de control de acceso (ACL).
- Instantáneas automáticas en Amazon S3 con retención de hasta 35 días.
- Support para hasta 500 nodos y más de 100 TB de almacenamiento por clúster (con 1 réplica por partición).
- Cifrado en tránsito con TLS y cifrado en reposo con claves de AWS KMS.
- Autenticación y autorización de usuarios con Redis Autenticación de usuarios con listas de control de acceso (ACL).
- Support para tipos de instancias de AWS Graviton2.
- Integración con otros AWS servicios como CloudWatch, Amazon VPC, CloudTrail y Amazon SNS para la supervisión, la seguridad y las notificaciones.
- Actualizaciones y parches de software totalmente gestionados.
- AWSIntegración de Identity and Access Management (IAM) y control de acceso basado en etiquetas para API de administración.

Componentes principales de MemoryDB

A continuación encontrará información general sobre los componentes principales de una implementación de MemoryDB.

Temas

- Clústeres
- Nodos
- Particiones
- Grupos de parámetros
- Grupos de subredes
- Listas de control de acceso
- Usuarios

Clústeres

Un clúster es una colección de uno o varios nodos que sirven a un conjunto de datos único. Un conjunto de datos de MemoryDB se divide en particiones y cada partición tiene un nodo principal y hasta 5 nodos de réplica opcionales. Un nodo principal atiende solicitudes de lectura y escritura, mientras que una réplica solo atiende solicitudes de lectura. Un nodo principal puede realizar una conmutación por error a un nodo de réplica, lo que permite pasar esa réplica al nuevo nodo principal de esa partición. MemoryDB ejecuta Redis como motor de base de datos y, cuando se crea un clúster, se especifica la versión de Redis del clúster. Puede modificar o crear un clúster de base de datos utilizando la AWS CLI, la o la API de MemoryDB o la AWS Management Console.

Cada clúster de MemoryDB ejecuta una versión del motor de Redis. Cada versión del motor de Redis tiene sus propias características compatibles. Además, cada versión del motor de Redis tiene un conjunto de parámetros en un grupo de parámetros que controla el comportamiento de los clústeres que administra.

La capacidad de cómputo y de memoria de un clúster se determina mediante su tipo de nodo. Puede seleccionar el tipo de nodo que mejor se adapte a sus necesidades. Si sus necesidades cambian con el tiempo, puede cambiar los tipos de nodo. Para obtener información, consulte Tipos de nodos compatibles.

> **Note**
>
> Para obtener información sobre los precios de los tipos de nodos de MemoryDB, consulte Precios de MemoryDB.

El clúster se ejecuta en una nube privada virtual (VPC) mediante el servicio Amazon Virtual Private Cloud (Amazon VPC). Cuando utilice una VPC, puede controlar todos los aspectos del entorno de

red virtual. Puede elegir su propio rango de direcciones IP, crear subredes y configurar listas de enrutamiento y control de acceso. MemoryDB administra las instantáneas, la aplicación de parches de software, la detección automática de errores y la recuperación. Es posible ejecutar el clúster en una VPC sin costo adicional. Para obtener más información acerca del uso de Amazon VPC con MemoryDB, consulte MemoryDB y Amazon VPC.

Muchas operaciones de MemoryDB están destinadas a los clústeres:

- creación de un clúster
- Modificación de un clúster
- Tomar instantáneas de un clúster
- Eliminación de un clúster
- Visualización de elementos de un clúster
- Adición o eliminación de etiquetas de asignación de costos en un clúster

Para obtener información más detallada, consulte los siguientes temas relacionados:

- Administración de clústeres y Administración de nodos

  Información acerca de los clústeres, nodos, y operaciones relacionadas.
- Resiliencia en MemoryDB para Redis

  Información sobre la mejora de la tolerancia a errores de los clústeres.

Nodos

Un nodo es el componente básico más pequeño de una implementación de MemoryDB y se ejecuta mediante una instancia de Amazon EC2. Cada nodo ejecuta la versión de Redis que se eligió al crear el clúster. Un nodo pertenece a una partición que pertenece a un clúster.

Cada nodo ejecuta una instancia del motor con la versión elegida al crear el clúster. Si es necesario, puede escalar o reducir verticalmente los nodos de un clúster a un tipo diferente. Para obtener más información, consulte Escalado .

Todos los nodos contenidos en un clúster son del mismo tipo. Se admiten varios tipos de nodos, cada uno con cantidades diferentes de memoria. Para ver una lista de los tipos de nodos admitidos, consulte Tipos de nodos compatibles.

Para obtener más información sobre los nodos, consulte Administración de nodos.

Particiones

Una partición es una agrupación de uno a 6 nodos, uno de los cuales actúa como nodo de escritura principal y los otros 5 como réplicas de lectura. Un clúster de MemoryDB siempre tiene al menos una partición.

Los clústeres de MemoryDB pueden tener hasta 500 particiones, con sus datos particionados en las particiones. Por ejemplo, puede elegir configurar un clúster de 500 nodos que oscila entre 83 particiones (uno primario y 5 réplicas por partición) y 500 particiones (único primario y sin réplicas). Asegúrese de que hay suficientes direcciones IP disponibles para acomodar el aumento. Algunos problemas comunes incluyen que las subredes del grupo de subredes tienen un rango CIDR demasiado pequeño o que otros clústeres comparten y utilizan considerablemente las subredes.

Una partición de varios nodos implementa la reproducción al tener un nodo principal de lectura/escritura y 1 a 5 nodos de réplica. Para obtener más información, consulte Descripción de cómo replicar en MemoryDB.

Para obtener más información acerca de las particiones, consulte Trabajar con particiones.

Grupos de parámetros

Los grupos de parámetros son una forma sencilla de administrar la configuración del tiempo de ejecución de Redis en su clúster. Los parámetros se utilizan para controlar el uso de la memoria, los tamaños de elementos y mucho más. Un grupo de parámetros de MemoryDB es un conjunto denominado de parámetros específicos del motor que se pueden aplicar a un clúster y todos los nodos de ese clúster se configuran exactamente de la misma forma.

Para obtener información más detallada acerca de los grupos de parámetros de MemoryDB, consulte Configuración de los parámetros de motor mediante los grupos de parámetros.

Grupos de subredes

Un grupo de subredes es una colección de subredes (que suelen ser privadas) que puede designar para los clústeres que se ejecutan en un entorno de Amazon Virtual Private Cloud (VPC).

Al crear un clúster en una Amazon VPC, pueden especificar un grupo de subredes o utilizar el grupo predeterminado que se proporciona. MemoryDB usa dicho grupo de subredes para elegir una subred y direcciones IP pertenecientes a dicha subred para asociarlas a sus nodos.

Para obtener información más detallada sobre los grupos de subredes de MemoryDB, consulte Subredes y grupos de subredes.

Listas de control de acceso

Una lista de control de acceso es un conjunto de uno o más usuarios. Las cadenas de acceso siguen las reglas de ACL de Redis para autorizar el acceso de los usuarios a los comandos y datos de Redis.

Para obtener información más detallada sobre las listas de control de acceso de MemoryDB, consulte Autenticación de usuarios con listas de control de acceso (ACL).

Usuarios

Un usuario tiene un nombre de usuario y una contraseña, y se utiliza para acceder a los datos y emitir comandos en su clúster de MemoryDB. Un usuario es miembro de una lista de control de acceso (ACL), que puede usar para determinar los permisos de ese usuario en los clústeres de MemoryDB. Para obtener más información, consultar Autenticación de usuarios con listas de control de acceso (ACL)

Servicios de relacionados

ElastiCache para Redis

Al decidir si utilizar MemoryDB para Redis o ElastiCache para Redis, tenga en cuenta las siguientes comparaciones:

- MemoryDB para Redis es una base de datos en memoria duradera para cargas de trabajo que requieren una base de datos principal ultrarrápida. Debería considerar el uso de MemoryDB si la carga de trabajo requiere una base de datos duradera que ofrezca un rendimiento ultrarrápido (lectura en microsegundos y latencia de escritura de un solo dígito en milisegundos). MemoryDB también puede ser una buena opción para el caso de uso si desea crear una aplicación mediante las estructuras de datos y las API de Redis con una base de datos principal y duradera. Por último, debería considerar el uso de MemoryDB para simplificar la arquitectura de la aplicación y reducir los costos al sustituir el uso de una base de datos por una memoria caché para aumentar la durabilidad y el rendimiento.
- ElastiCache para Redis es un servicio que se utiliza normalmente para almacenar en caché datos de otras bases de datos y almacenes de datos con Redis. Debería considerar ElastiCache para

Redis para almacenar en caché cargas de trabajo en las que desee acelerar el acceso a los datos con la base de datos principal o almacén de datos existente (rendimiento de lectura y escritura en microsegundos). También debe considerar ElastiCache para Redis para los casos de uso en los que desee utilizar las estructuras de datos y las API de Redis para acceder a los datos almacenados en una base de datos o un banco de datos principal.

Elección de regiones y zonas de disponibilidad

AWSLos recursos de informática en la nube de se alojan en centros de datos de alta disponibilidad. Para proporcionar escalabilidad y fiabilidad adicionales, estas instalaciones de centros de datos se encuentran en ubicaciones físicas diferentes. Dichas ubicaciones están categorizadas por regiones y zonas de disponibilidad.

Las regiones de AWS son de gran tamaño y se encuentran dispersas en distintas ubicaciones geográficas. Las zonas de disponibilidad son ubicaciones concretas dentro de una región de AWS diseñadas para estar aisladas de los errores que se produzcan en las demás zonas de disponibilidad. Proporcionan conectividad de red económica y de baja latencia con las demás zonas de disponibilidad dentro de la misma región de AWS.

> **Important**
>
> Cada región es totalmente independiente. Cualquier actividad de MemoryDB que inicie (por ejemplo, la creación de clústeres) solo se ejecutará en la región predeterminada actual.

Para crear o trabajar con un clúster de una región específica, use el punto de conexión de servicio regional correspondiente. Para obtener información acerca de los puntos de conexión del servicio, consulte Regiones y puntos de conexión admitidos.

Ubicación de los nodos

Cualquier clúster que tenga al menos una réplica debe estar distribuido entre las AZ. La única forma de localizar todo lo que hay dentro de una única zona de disponibilidad es con un clúster compuesto por particiones de un solo nodo.

Al ubicar los nodos en distintas zonas de disponibilidad, MemoryDB elimina la posibilidad de que un fallo, como un corte del suministro eléctrico, en una zona de disponibilidad resulte en una falta de disponibilidad.

- Creación de un clúster de MemoryDB
- Modificación de un clúster de MemoryDB

Regiones y puntos de conexión admitidos

MemoryDB para Redis se encuentra disponible en varias regiones de AWS. Esto significa que puede lanzar clústeres de MemoryDB en las ubicaciones que cumplan sus requisitos. Por ejemplo, puede lanzarlos en la región de AWS más cercana a los clientes o en una región de AWS concreta que permita cumplir determinados requisitos legales. Además, cuando MemoryDB amplía su disponibilidad a una nueva región de AWS, MemoryDB admite las dos versiones `MAJOR.MINOR` más recientes en ese momento para la nueva región. Para obtener más información acerca de las versiones de MemoryDB, consulte Versiones del motor de Redis.

De forma predeterminada, los AWS SDKAWS CLI, la API de MemoryDB y la consola de MemoryDB hacen referencia a la región este de EE. UU. (Norte de Virginia). A medida que MemoryDB amplía la disponibilidad a regiones nuevas, los puntos de conexión nuevos de estas regiones también se encuentran disponibles para su uso en las solicitudes HTTP, los SDK de AWS, la AWS CLI y la consola.

Cada región de se ha diseñado para que se encuentre totalmente aislada de las demás regiones de . Dentro de cada región hay varias zonas de disponibilidad. Al lanzar los nodos en zonas de disponibilidad diferentes, puede lograr la máxima tolerancia a errores. Para obtener más información acerca de las regiones y zonas de disponibilidad, consulte Elección de regiones y zonas de disponibilidad al comienzo de este tema.

Regiones en las que se admite MemoryDB

Nombre de la región/ Región	Punto de conexión	Protocolo	
Región del este de EE. UU. (Ohio) `us-east-2`	`memory-db.us-east-2.amazonaws.com`	HTTPS	
Región del este de EE. UU (N. Virginia) `us-east-1`	`memory-db.us-east-1.amazonaws.com`	HTTPS	
Región del oeste de EE. UU (N. California)	`memory-db.us-west-1.amazonaws.com`	HTTPS	

Nombre de la región/ Región	Punto de conexión	Protocolo	
us-west-1			
Región del oeste de EE. UU (Oregon) us-west-2	memory-db.us-west-2.amazonaws.com	HTTPS	
Canada (Central) Region ca-central-1	memory-db.ca-central-1.amazonaws.com	HTTPS	
Región Asia Pacífico (Hong Kong) ap-east-1	memory-db.ap-eastl-1.amazonaws.com	HTTPS	
Región Asia-Pacífico (Mumbai) ap-south-1	memory-db.ap-south-1.amazonaws.com	HTTPS	
Región Asia-Pacífico (Tokio) ap-northeast-1	memory-db.ap-northeast-1.amazonaws.com	HTTPS	
Región Asia-Pacífico (Seúl) ap-northeast-2	memory-db.ap-northeast-2.amazonaws.com	HTTPS	
Región Asia-Pacífico (Singapur) ap-southeast-1	memory-db.ap-southeast-1.amazonaws.com	HTTPS	

Nombre de la región/ Región	Punto de conexión	Protocolo	
Región Asia-Pacífico (Sídney) `ap-southeast-2`	`memory-db.ap-southeast-2.amazonaws.com`	HTTPS	
Europe (Frankfurt) Region `eu-central-1`	`memory-db.eu-central-1.amazonaws.com`	HTTPS	
Europe (Ireland) Region `eu-west-1`	`memory-db.eu-west-1.amazonaws.com`	HTTPS	
Europe (London) Region `eu-west-2`	`memory-db.eu-west-2.amazonaws.com`	HTTPS	
Región EU (París) `eu-west-3`	`memory-db.eu-west-3.amazonaws.com`	HTTPS	
Región Europa (Estocolmo) `eu-north-1`	`memory-db.eu-north-1.amazonaws.com`	HTTPS	
Europe (Milan) Region `eu-south-1`	`memory-db.eu-south-1.amazonaws.com`	HTTPS	

Nombre de la región/ Región	Punto de conexión	Protocolo	
South America (São Paulo) Region `sa-east-1`	`memory-db.sa-east-1.amazonaws.com`	HTTPS	
Región China (Pekín) `cn-north-1`	`memory-db.cn-north-1.amazonaws.com.cn`	HTTPS	
Región China (Ningxia) `cn-northwest-1`	`memory-db.cn-northwest-1.amazonaws.com.cn`	HTTPS	

Para obtener una tabla de los productos y servicios de AWS por región, consulte Productos y servicios por región.

Para ver una tabla de las zonas de disponibilidad admitidas dentro de las regiones, consulte Subredes y grupos de subredes.

Acceso a MemoryDB

Cada punto de conexión del clúster de MemoryDB contiene una dirección y un puerto. Este punto de conexión del clúster es compatible con el protocolo Redis Cluster, que permite a los clientes descubrir los roles, las direcciones IP y las ranuras específicas de cada nodo del clúster. Cuando se produce un error en un nodo principal y se coloca una réplica en su lugar, puede conectarse al punto de conexión del clúster para detectar el nuevo nodo principal mediante el protocolo Redis Cluster.

Debe conectarse al punto de conexión del clúster para detectar los puntos de conexión del nodo mediante un comando cluster nodes o cluster slots. Tras encontrar el nodo correcto para una clave, puede conectarse directamente al nodo para realizar solicitudes de lectura/escritura. Un cliente de Redis puede usar el punto de conexión del clúster para conectarse automáticamente al nodo correcto.

Para solucionar problemas de nodos específicos de un clúster, también puede utilizar puntos de conexión específicos de cada nodo, pero no son necesarios para un uso normal.

Para encontrar el punto de conexión de un clúster, consulte lo siguiente:

- Búsqueda del punto de conexión para un clúster de MemoryDB (CLI de AWS)
- Búsqueda del punto de conexión para un clúster de MemoryDB (API de MemoryDB)

Para conectarse a nodos o clústeres, consulte Conexión a los nodos de MemoryDB mediante redis-cli.

Seguridad de MemoryDB

La seguridad de MemoryDB se administra en tres niveles:

- Para controlar quién puede realizar acciones de administración en clústeres y nodos de MemoryDB, se usa AWS Identity and Access Management (IAM). Cuando se conecta a AWS con credenciales de IAM, la cuenta de AWS debe tener políticas de IAM que concedan los permisos necesarios para realizar operaciones. Para obtener más información, consulte Administración de identidades y accesos en MemoryDB para Redis
- Para controlar los niveles de acceso a los clústeres, debe crear usuarios con permisos específicos y asignarlos a las listas de control de acceso (ACL). La ACL, a su vez, se asocia entonces a uno o más clústeres. Para obtener más información, consulte Autenticación de usuarios con listas de control de acceso (ACL).
- Los clústeres de base de datos de MemoryDB deben crearse en una nube privada virtual (VPC) basada en el servicio de Amazon VPC. Para controlar qué dispositivos e instancias de Amazon EC2 pueden abrir conexiones al punto de conexión y al puerto del nodo de los clústeres de MemoryDB en una VPC, debe usar un grupo de seguridad de VPC. Puede establecer estas conexiones de puerto y punto de conexión mediante Transport Layer Security (TLS)/Capa de conexión segura (SSL). Además, las reglas del firewall de su empresa pueden controlar si los dispositivos que se ejecutan en ella pueden abrir conexiones a un clúster de MemoryDB. Para obtener más información acerca de las VPC, consulte MemoryDB y Amazon VPC.

Para obtener información acerca de la configuración de seguridad, consulte Seguridad en MemoryDB para Redis.

Introducción a MemoryDB

Este ejercicio explica los pasos para crear, conceder acceso, conectarse y, finalmente, eliminar un clúster de MemoryDB mediante la consola de administración de MemoryDB.

> **Note**
>
> A los fines de este ejercicio, le recomendamos que utilice la opción Creación sencilla al crear un clúster y que vuelva a las otras dos opciones una vez que haya explorado más en detalle las funciones de MemoryDB.

Temas

Configuración

A continuación encontrará los temas que describen las acciones puntuales que es preciso realizar para comenzar a usar MemoryDB.

Temas

Crear la cuenta de AWS

Registro para obtener una Cuenta de AWS

Si no dispone de una Cuenta de AWS, siga estos pasos para crear una.

Cómo registrarse en una Cuenta de AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

   Al registrarse en una Cuenta de AWS, se crea un Usuario raíz de la cuenta de AWS. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario administrativo y utilice únicamente el usuario raíz para realizar tareas que requieran acceso de usuario raíz.

AWS le enviará un correo electrónico de confirmación luego de completar el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en https://aws.amazon.com/ y seleccionando Mi cuenta.

Crear un usuario administrativo

Después de registrarse para obtener una Cuenta de AWS, proteja su Usuario raíz de la cuenta de AWS, habilite AWS IAM Identity Center y cree un usuario administrativo para no utilizar el usuario raíz en las tareas cotidianas.

Protección de su Usuario raíz de la cuenta de AWS

1. Inicie sesión en la AWS Management Console como propietario de cuenta, elija Usuario raíz e ingrese el email de su Cuenta de AWS. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Iniciar sesión como usuario raíz en la Guía del usuario de AWS Sign-In.
2. Active la autenticación multifactor (MFA) para el usuario raíz.

   Para obtener instrucciones, consulte Habilitar un dispositivo MFA virtual para el usuario raíz de la Cuenta de AWS (consola) en la Guía del usuario de IAM.

Creación de un usuario administrativo

1. Activar IAM Identity Center

   Consulte las instrucciones en Enabling AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center.

2. En IAM Identity Center, conceda acceso administrativo a un usuario administrativo.

   Para ver un tutorial sobre el uso del Directorio de IAM Identity Center como origen de identidades, consulte Configuración del acceso de los usuarios con el Directorio de IAM Identity Center predeterminado en la Guía del usuario de AWS IAM Identity Center.

Cómo iniciar sesión como usuario administrativo

- Para iniciar sesión con el usuario del IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario del IAM Identity Center.

  Para obtener ayuda para iniciar sesión con un usuario del IAM Identity Center, consulte Iniciar sesión en el portal de acceso de AWS en la Guía del Usuario de AWS Sign-In.

Conceder acceso programático

Los usuarios necesitan acceso programático si desean interactuar con AWS fuera de la AWS Management Console. La forma de conceder el acceso programático depende del tipo de usuario que acceda a AWS.

Para conceder acceso programático a los usuarios, seleccione una de las siguientes opciones.

¿Qué usuario necesita acceso programático?	Para	Mediante
Identidad del personal (Usuarios administrados en el Centro de identidades de IAM)	Utilice credenciales temporales para firmar las solicitudes programáticas a la AWS CLI, los SDK de AWS o las API de AWS.	Siga las instrucciones de la interfaz que desea utilizar: • En el caso de la AWS CLI, consulte Configuración de la AWS CLI para usar AWS IAM Identity Center en la

¿Qué usuario necesita acceso programático?	Para	Mediante
		Guía del usuario de AWS Command Line Interface. • En el caso de los SKD de AWS, las herramientas y las API de AWS, consulte Autenticación de IAM Identity Center en la Guía de referencia de SDK de AWS y herramientas.
IAM	Utilice credenciales temporales para firmar las solicitudes programáticas a la AWS CLI, los SDK de AWS o las API de AWS.	Siga las instrucciones de Uso de credenciales temporales con recursos de AWS de la Guía del Usuario de IAM.

¿Qué usuario necesita acceso programático?	Para	Mediante
IAM	(no recomendado) Utilice credenciales a largo plazo para firmar las solicitud es programáticas a la AWS CLI, los SDK de AWS o las API de AWS.	Siga las instrucciones de la interfaz que desea utilizar: • Para la AWS CLI, consulte Autenticación mediante credenciales de usuario de IAM en la Guía del usuario de AWS Command Line Interface. • Para los SDK de AWS y las herramientas, consulte Autenticación mediante credenciales a largo plazo en la Guía de referencia de SDK de AWS y herramien tas. • Para las API de AWS, consulte Administración de claves de acceso para usuarios de IAM en la Guía del usuario de IAM.

Temas relacionados:

- ¿Qué es IAM? en la Guía del usuario de IAM.
- Credenciales de seguridad de AWS en la Referencia general de AWS.

Configurar los permisos (solo para nuevos usuarios de MemoryDB)

Para dar acceso, añada permisos a los usuarios, grupos o roles:

- Usuarios y grupos en AWS IAM Identity Center:

Cree un conjunto de permisos. Siga las instrucciones de Creación de un conjunto de permisos en la Guía del usuario de AWS IAM Identity Center.

- Usuarios administrados en IAM a través de un proveedor de identidades:

  Cree un rol para la federación de identidades. Siga las instrucciones de Creación de un rol para un proveedor de identidades de terceros (federación) en la Guía del usuario de IAM.
- Usuarios de IAM:
  - Cree un rol que el usuario pueda aceptar. Siga las instrucciones descritas en Creación de un rol para un usuario de IAM en la Guía del usuario de IAM.
  - (No recomendado) Adjunte una política directamente a un usuario o añada un usuario a un grupo de usuarios. Siga las instrucciones descritas en Adición de permisos a un usuario (consola) de la Guía del usuario de IAM.

MemoryDB para Redis crea y utiliza roles vinculados a servicios para aprovisionar recursos y obtener acceso a otros servicios y recursos de AWS en su nombre. Para que MemoryDB cree un rol vinculado a un servicio para usted, utilice la política administrada por AWS denominada `AmazonMemoryDBFullAccess`. Esta función ya está aprovisionada con los permisos que el servicio requiere para crear un rol vinculado a un servicio en su nombre.

Si lo prefiere, puede no utilizar la política predeterminada, sino una administrada de forma personalizada. En este caso, asegúrese de tener los permisos para llamar a `iam:createServiceLinkedRole` o bien de haber creado el rol vinculado al servicio de MemoryDB.

Para obtener más información, consulte los siguientes temas:

- Creación de una nueva política (IAM)
- Políticas (predefinidas) administradas por AWS para MemoryDB para Redis
- Uso de roles vinculados a servicios para Amazon MemoryDB para Redis

Descarga y configuración de la AWS CLI

La AWS CLI se encuentra disponible en http://aws.amazon.com/cli. Se ejecuta en Windows, MacOS y Linux. Después de descargar la AWS CLI, siga estos pasos para instalarla y configurarla:

1. Diríjase a la Guía del usuario de la interfaz de la línea de comandos de AWS.

2. Siga las instrucciones para Instalar la AWS CLI y Configurar la AWS CLI.

Paso 1: creación de un clúster

Antes de crear un clúster para su uso en producción, obviamente debe considerar cómo configurará el clúster a fin de satisfacer las necesidades del negocio. Estos problemas se abordan en la sección Preparación de un clúster. A los efectos de este ejercicio de introducción, puede aceptar los valores de configuración predeterminados donde se apliquen.

El clúster que crea se ejecutará en un entorno real, no en uno de pruebas. Deberá pagar las tarifas de uso estándares de MemoryDB para la instancia hasta que la elimine. Los cargos totales serán mínimos (normalmente menos de un dólar) si completa el ejercicio descrito aquí de una vez y elimina el clúster al finalizar. Para obtener más información sobre las tarifas de uso de MemoryDB, consulte MemoryDB.

El clúster se lanza en una nube privada virtual (VPC) en función del servicio de Amazon VPC.

Creación de un clúster de MemoryDB

Los siguientes ejemplos muestran cómo crear un clúster mediante la API AWS Management Console, AWS CLI y MemoryDB.

Creación de un clúster (consola)

Para crear un clúster utilizando la consola de MemoryDB

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. Seleccione Clústeres en el panel de navegación izquierdo y, a continuación, seleccione Crear.

Easy create

1. Rellene la sección Configuration (Configuración). Esta acción configura el tipo de nodo y la configuración predeterminada del clúster. Seleccione el tamaño de la memoria y el rendimiento de red adecuados según sus necesidades de entre las siguientes opciones:
   - Producción
   - Desarrollo/pruebas
   - Demostración
2. Complete la sección de información del clúster.

a. En Nombre, escriba un nombre para su clúster.

   Las restricciones para la asignación de nombres de clúster son las siguientes:

   - Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
   - Deben comenzar por una letra.
   - No pueden contener dos guiones consecutivos.
   - No pueden terminar con un guion.

b. En el cuadro Descripción, especifique una descripción para este clúster.

3. Complete la sección Grupos de subredes:

   - En Grupos de subredes, cree un nuevo grupo de subredes o elija uno existente de la lista disponible que desee aplicar a este clúster. Si va a crear uno nuevo:

     - Escriba un nombre
     - Escriba una descripción
     - Si ha habilitado Multi-AZ, el grupo de subredes debe contener al menos dos subredes que residan en zonas de disponibilidad diferentes. Para obtener más información, consulte Subredes y grupos de subredes.
     - Si va a crear un nuevo grupo de subredes y no tiene una VPC existente, se le pedirá que cree una VPC. Para obtener más información, consulte ¿Qué es Amazon VPC? en la Guía del usuario de Amazon VPC.

4. En Búsqueda vectorial, puede Habilitar la capacidad de búsqueda vectorial para almacenar incrustaciones vectoriales y realizar búsquedas vectoriales. Tenga en cuenta que esto fijará los valores de Compatibilidad con versión de Redis, Grupos de parámetros y Particiones. Para obtener más información, consulte Búsqueda vectorial.

5. Ver la configuración predeterminada:

   Cuando se utiliza Creación sencilla, el resto de la configuración del clúster se establece de forma predeterminada. Tenga en cuenta que algunos de estos ajustes se pueden cambiar después de la creación, tal y como se indica en Editable tras la creación.

6. En el caso de las etiquetas, si lo desea, puede aplicar etiquetas para buscar y filtrar sus clústeres o realizar un seguimiento de sus costes. AWS

7. Revise todas las entradas y opciones y, a continuación, realice todos los cambios necesarios. Cuando esté listo, elija Crear clúster para lanzar su clúster, o bien Cancelar para cancelar la operación.

En cuanto el estado de su clúster sea available, podrá conceder a EC2 acceso a este, así como conectarse a él y comenzar a utilizarlo. Para más información, consulte Paso 2: autorizar el acceso al clúster

> **Important**
>
> Cuando su clúster esté disponible, se cobrará por cada hora u hora parcial que el clúster esté activo, incluso si no lo está utilizando de forma activa. Para dejar de incurrir en cargos por este clúster, debe eliminarlo. Consulte Paso 4: eliminar un clúster.

Create new cluster

1. Complete la sección de información del clúster.
   a. En Nombre, escriba un nombre para su clúster.

      Las restricciones para la asignación de nombres de clúster son las siguientes:

      - Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
      - Deben comenzar por una letra.
      - No pueden contener dos guiones consecutivos.
      - No pueden terminar con un guion.

   b. En el cuadro Descripción, especifique una descripción para este clúster.
2. Complete la sección Grupos de subredes:
   - En Grupos de subredes, cree un nuevo grupo de subredes o elija uno existente de la lista disponible que desee aplicar a este clúster. Si va a crear uno nuevo:
     - Escriba un nombre
     - Escriba una descripción

- Si ha habilitado Multi-AZ, el grupo de subredes debe contener al menos dos subredes que residan en zonas de disponibilidad diferentes. Para obtener más información, consulte Subredes y grupos de subredes.
- Si va a crear un nuevo grupo de subredes y no tiene una VPC existente, se le pedirá que cree una VPC. Para obtener más información, consulte ¿Qué es Amazon VPC? en la Guía del usuario de Amazon VPC.

3. Complete la sección Configuración del clúster:

   a. En Habilitar la capacidad de búsqueda vectorial, puede habilitarla para almacenar incrustaciones vectoriales y realizar búsquedas vectoriales. Tenga en cuenta que esto fijará los valores de Compatibilidad con versión de Redis, Grupos de parámetros y Particiones. Para obtener más información, consulte Búsqueda vectorial.

   b. En Compatibilidad con versión de Redis, acepte la versión de `6.2` predeterminada.

   c. En Port, acepte 6379 como puerto predeterminado de Redis o, si tiene algún motivo para utilizar un puerto diferente, introduzca el número de puerto.

   d. En Grupo de parámetros, si ha habilitado la búsqueda vectorial, utilice `default.memorydb-redis7.search.preview`. De lo contrario, acepte el grupo de parámetros `default.memorydb-redis7`.

      Los grupos de parámetros controlan los parámetros de tiempo de ejecución de su clúster. Para obtener más información acerca de los grupos de parámetros, consulte Parámetros específicos de Redis.

   e. En Tipo de nodo, elija un valor para el tipo de nodo (junto con el tamaño de memoria asociado) que desee.

      Si elige un tipo de nodo de la familia r6gd, habilitará automáticamente la organización de datos en niveles, que divide el almacenamiento de datos entre la memoria y la SSD. Para obtener más información, consulte Organización de datos en niveles.

   f. En Número de particiones, elija el número de particiones que desea para este clúster. Para aumentar la disponibilidad de sus clústeres, le recomendamos que añada al menos 2 particiones.

      Puede cambiar dinámicamente el número de particiones del clúster. Para obtener más información, consulte Escalado de clústeres de MemoryDB.

g. En Réplicas por partición, elija el número de nodos de réplica de lectura que desea en cada partición.

   Existen las siguientes restricciones:

   - Si tiene habilitado Multi-AZ, asegúrese de tener al menos una réplica por partición.
   - El número de réplicas es el mismo para cada fragmento al crear el clúster utilizando la consola.

h. Elija Siguiente.

i. Complete la sección Configuración avanzada:

   i. En Grupos de seguridad, elija los grupos de seguridad que desea para este clúster. Un grupo de seguridad actúa como un firewall para controlar el acceso de red al clúster. Puede utilizar el grupo de seguridad predeterminado para la VPC o crear uno nuevo.

      Para obtener más información sobre los grupos de seguridad, consulte Grupos de seguridad de su VPC en la Guía del usuario de Amazon VPC.

   ii. Para cifrar sus datos, tiene las siguientes opciones:

      - Encryption at rest (Cifrado en reposo): permite el cifrado de los datos almacenados en el disco. Para obtener más información, consulte Cifrado en reposo.

        > **Note**
        >
        > Tiene la opción de proporcionar una clave de cifrado distinta de la predeterminada. Para ello, seleccione la clave KMS AWS propiedad de Customer Managed y elija la clave.

      - Encryption in-transit (Cifrado en tránsito): permite el cifrado de datos del cable. Si no selecciona ningún cifrado, se creará una lista de control de acceso abierta denominada “acceso abierto” con un usuario predeterminado. Para obtener más información, consulte Autenticación de usuarios con listas de control de acceso (ACL).

iii. En el caso de una instantánea, si lo desea, especifique un periodo de retención de instantáneas y un periodo de instantáneas. De forma predeterminada, la opción Habilitar instantáneas automáticas está preseleccionada.

iv. En el periodo de mantenimiento, especifique opcionalmente un periodo de mantenimiento. El periodo de mantenimiento es el tiempo, generalmente de una hora, de cada semana durante el que MemoryDB programa el mantenimiento del sistema para su clúster. Puede permitir que MemoryDB elija el día y la hora de su periodo de mantenimiento (Sin preferencia) o bien puede elegir el día, la hora y la duración por su cuenta (Especificar periodo de mantenimiento). Si elige Specify maintenance window, elija Start day, Start time y Duration (en horas) de las listas para el periodo de mantenimiento. Todas las horas se indican en UCT.

Para obtener más información, consulte Administración del mantenimiento.

v. En Notifications (Notificaciones), elija un tema existente de Amazon Simple Notification Service (Amazon SNS) o bien una entrada de ARN manual y escriba el tema nombre de recurso de Amazon (ARN). Amazon SNS le permite enviar notificaciones de inserción a dispositivos inteligentes con conexión a Internet. El valor predeterminado tiene las notificaciones deshabilitadas. Para obtener más información, consulte https://aws.amazon.com/sns/.

vi. En el caso de las etiquetas, si lo desea, puede aplicar etiquetas para buscar y filtrar sus clústeres o realizar un seguimiento de sus AWS costes.

j. Revise todas las entradas y opciones y, a continuación, realice todos los cambios necesarios. Cuando esté listo, elija Crear clúster para lanzar su clúster, o bien Cancelar para cancelar la operación.

En cuanto el estado de su clúster sea available, podrá conceder a EC2 acceso a este, así como conectarse a él y comenzar a utilizarlo. Para más información, consulte Paso 2: autorizar el acceso al clúster

> **Important**
>
> Cuando su clúster esté disponible, se cobrará por cada hora u hora parcial que el clúster esté activo, incluso si no lo está utilizando de forma activa. Para dejar de incurrir en cargos por este clúster, debe eliminarlo. Consulte Paso 4: eliminar un clúster.

Restore from snapshots

En Origen de la instantánea, elija la instantánea de origen desde donde migrar los datos. Para obtener más información, consulte Instantánea y restauración .

> **Note**
>
> Si quiere que su nuevo clúster tenga habilitada la búsqueda vectorial, la instantánea de origen también debe tener habilitada la búsqueda vectorial.

El clúster de destino usa de forma predeterminada la configuración del clúster de origen. Si lo prefiere, puede cambiar la siguiente configuración en el clúster de destino:

1. Información del clúster

   a. En Nombre, escriba un nombre para su clúster.

      Las restricciones para la asignación de nombres de clúster son las siguientes:

      - Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
      - Deben comenzar por una letra.
      - No pueden contener dos guiones consecutivos.
      - No pueden terminar con un guion.

   b. En el cuadro Descripción, especifique una descripción para este clúster.

2. Grupos de subredes

   - En Grupos de subredes, cree un nuevo grupo de subredes o elija uno existente de la lista disponible que desee aplicar a este clúster. Si va a crear uno nuevo:

     - Escriba un nombre
     - Escriba una descripción
     - Si ha habilitado Multi-AZ, el grupo de subredes debe contener al menos dos subredes que residan en zonas de disponibilidad diferentes. Para obtener más información, consulte Subredes y grupos de subredes.
     - Si va a crear un nuevo grupo de subredes y no tiene una VPC existente, se le pedirá que cree una VPC. Para obtener más información, consulte ¿Qué es Amazon VPC? en la Guía del usuario de Amazon VPC.

3. Configuración del clúster

   a. En Habilitar la capacidad de búsqueda vectorial, puede habilitarla para almacenar incrustaciones vectoriales y realizar búsquedas vectoriales. Tenga en cuenta que esto fijará los valores de Compatibilidad con versión de Redis, Grupos de parámetros y Particiones. Para obtener más información, consulte Búsqueda vectorial.

   b. En Compatibilidad con versión de Redis, acepte la versión de `6.2` predeterminada.

   c. En Port, acepte 6379 como puerto predeterminado de Redis o, si tiene algún motivo para utilizar un puerto diferente, introduzca el número de puerto.

   d. En Grupo de parámetros, si ha habilitado la búsqueda vectorial, utilice `default.memorydb-redis7.search.preview`. De lo contrario, acepte el grupo de parámetros `default.memorydb-redis7`.

      Los grupos de parámetros controlan los parámetros de tiempo de ejecución de su clúster. Para obtener más información acerca de los grupos de parámetros, consulte Parámetros específicos de Redis.

   e. En Tipo de nodo, elija un valor para el tipo de nodo (junto con el tamaño de memoria asociado) que desee.

      Si elige un tipo de nodo de la familia r6gd, habilitará automáticamente la organización de datos en niveles, que divide el almacenamiento de datos entre la memoria y la SSD. Para obtener más información, consulte Organización de datos en niveles.

   f. En Número de particiones, elija el número de particiones que desea para este clúster. Para aumentar la disponibilidad de sus clústeres, le recomendamos que añada al menos 2 particiones.

      Puede cambiar dinámicamente el número de particiones del clúster. Para obtener más información, consulte Escalado de clústeres de MemoryDB.

   g. En Réplicas por partición, elija el número de nodos de réplica de lectura que desea en cada partición.

      Existen las siguientes restricciones:

      - Si tiene habilitado Multi-AZ, asegúrese de tener al menos una réplica por partición.
      - El número de réplicas es el mismo para cada fragmento al crear el clúster utilizando la consola.

h. Elija Siguiente.

i. Configuración avanzada

i. En Grupos de seguridad, elija los grupos de seguridad que desea para este clúster. Un grupo de seguridad actúa como un firewall para controlar el acceso de red al clúster. Puede utilizar el grupo de seguridad predeterminado para la VPC o crear uno nuevo.

Para obtener más información sobre los grupos de seguridad, consulte Grupos de seguridad de su VPC en la Guía del usuario de Amazon VPC.

ii. Para cifrar sus datos, tiene las siguientes opciones:

- Encryption at rest (Cifrado en reposo): permite el cifrado de los datos almacenados en el disco. Para obtener más información, consulte Cifrado en reposo.

> **Note**
>
> Tiene la opción de proporcionar una clave de cifrado distinta de la predeterminada. Para ello, seleccione la clave KMS AWS propiedad de Customer Managed y elija la clave.

- Encryption in-transit (Cifrado en tránsito): permite el cifrado de datos del cable. Si no selecciona ningún cifrado, se creará una lista de control de acceso abierta denominada “acceso abierto” con un usuario predeterminado. Para obtener más información, consulte Autenticación de usuarios con listas de control de acceso (ACL).

iii. En el caso de una instantánea, si lo desea, especifique un periodo de retención de instantáneas y un periodo de instantáneas. De forma predeterminada, la opción Habilitar instantáneas automáticas está preseleccionada.

iv. En el periodo de mantenimiento, especifique opcionalmente un periodo de mantenimiento. El periodo de mantenimiento es el tiempo, generalmente de una hora, de cada semana durante el que MemoryDB programa el mantenimiento del sistema para su clúster. Puede permitir que MemoryDB elija el día y la hora de su periodo de mantenimiento (Sin preferencia) o bien puede elegir el día, la hora y la duración por su cuenta (Especificar periodo de mantenimiento). Si elige Specify

maintenance window, elija Start day, Start time y Duration (en horas) de las listas para el periodo de mantenimiento. Todas las horas se indican en UCT.

Para obtener más información, consulte Administración del mantenimiento.

v. En Notifications (Notificaciones), elija un tema existente de Amazon Simple Notification Service (Amazon SNS) o bien una entrada de ARN manual y escriba el tema nombre de recurso de Amazon (ARN). Amazon SNS le permite enviar notificaciones de inserción a dispositivos inteligentes con conexión a Internet. El valor predeterminado tiene las notificaciones deshabilitadas. Para obtener más información, consulte https://aws.amazon.com/sns/.

vi. En el caso de las etiquetas, si lo desea, puede aplicar etiquetas para buscar y filtrar sus clústeres o realizar un seguimiento de sus AWS costes.

j. Revise todas las entradas y opciones y, a continuación, realice todos los cambios necesarios. Cuando esté listo, elija Crear clúster para lanzar su clúster, o bien Cancelar para cancelar la operación.

En cuanto el estado de su clúster sea available, podrá conceder a EC2 acceso a este, así como conectarse a él y comenzar a utilizarlo. Para más información, consulte Paso 2: autorizar el acceso al clúster

> **Important**
>
> Cuando su clúster esté disponible, se cobrará por cada hora u hora parcial que el clúster esté activo, incluso si no lo está utilizando de forma activa. Para dejar de incurrir en cargos por este clúster, debe eliminarlo. Consulte Paso 4: eliminar un clúster.

Creación de un clúster (AWS CLI)

Para crear un clúster mediante AWS CLI, consulte `create-cluster`. A continuación se muestra un ejemplo:

Para Linux, macOS o Unix:

```
aws memorydb create-cluster \
    --cluster-name my-cluster \
    --node-type db.r6g.large \
    --acl-name my-acl \
    --subnet-group my-sg
```

Para Windows:

```
aws memorydb create-cluster ^
   --cluster-name my-cluster ^
   --node-type db.r6g.large ^
   --acl-name my-acl ^
   --subnet-group my-sg
```

Debería obtener la siguiente respuesta JSON:

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "creating",
        "NumberOfShards": 1,
        "AvailabilityMode": "MultiAZ",
        "ClusterEndpoint": {
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxxxxxxxxx:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
```

```
        "SnapshotWindow": "04:30-05:30",
        "ACLName": "my-acl",
        "DataTiering": "false",
        "AutoMinorVersionUpgrade": true
    }
}
```

Puede empezar a usar el clúster una vez que su estado cambie a `available`.

> **Important**
>
> Cuando su clúster esté disponible, se cobrará por cada hora u hora parcial que el clúster esté activo, incluso si no lo está utilizando de forma activa. Para dejar de incurrir en cargos por este clúster, debe eliminarlo. Consulte Paso 4: eliminar un clúster.

Creación de un clúster (API de MemoryDB)

Para crear un clúster mediante la API MemoryDB, usa la CreateClusteracción.

> **Important**
>
> Cuando su clúster esté disponible, se le cobrará por cada hora u hora parcial que el clúster esté activo, incluso si no lo está utilizando. Para dejar de incurrir en cargos por este clúster, debe eliminarlo. Consulte Paso 4: eliminar un clúster.

Configuración de la autenticación

Para obtener información sobre cómo configurar la autenticación para el clúster, consulte Autenticación con IAM yAutenticación de usuarios con listas de control de acceso (ACL).

Paso 2: autorizar el acceso al clúster

En esta sección se presupone que se encuentra familiarizado con los procesos de lanzar instancias de Amazon EC2 y conectarse a estas. Para obtener más información, consulte la Guía de introducción de Amazon EC2.

Los clústeres de MemoryDB se han diseñado para que se pueda obtener acceso a ellos desde una instancia de Amazon EC2. También se puede acceder a ellos mediante aplicaciones en contenedores o sin servidor que se ejecuten en Amazon Elastic Container Service o AWS Lambda. Lo más habitual es obtener acceso a un clúster de MemoryDB desde una instancia de Amazon EC2 en la misma Amazon Virtual Private Cloud (Amazon VPC); este será el caso utilizaremos para este ejercicio.

Antes de poder conectarse a un clúster desde una instancia EC2, debe autorizar a la instancia EC2 el acceso al clúster.

El caso de uso más común es cuando una aplicación implementada en una instancia EC2 debe conectarse a un clúster en la misma VPC. La forma más sencilla de administrar el acceso entre instancias EC2 y clústeres de la misma VPC es realizar lo siguiente:

1. Cree un grupo de seguridad de VPC para su clúster. Este grupo de seguridad se puede utilizar para restringir el acceso a los clústeres. Por ejemplo, puede crear una regla personalizada para este grupo de seguridad que permita el acceso mediante TCP utilizando el puerto que asignó al clúster de base de datos cuando lo creó y una dirección IP que se utilizará para obtener acceso al clúster.

   El puerto predeterminado para los clústeres de MemoryDB es 6379.
2. Cree un grupo de seguridad de VPC para sus instancias EC2 (servidores web y de aplicaciones). Si es necesario, este grupo de seguridad puede permitir el acceso a la instancia EC2 desde Internet a través de la tabla de enrutamiento de la VPC. Por ejemplo, puede establecer reglas en este grupo de seguridad para permitir el acceso mediante TCP a la instancia EC2 a través del puerto 22.
3. Cree reglas personalizadas en el grupo de seguridad para su clúster que permitan las conexiones desde el grupo de seguridad que creó para las instancias EC2. Esto permitirá a cualquier miembro del grupo de seguridad obtener acceso a los clústeres.

Para crear una regla en un grupo de seguridad de VPC que permita establecer conexiones desde otro grupo de seguridad

1. Inicie sesión en la consola AWS de administración y abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc.
2. En el panel de navegación izquierdo, elija Security Groups.
3. Seleccione o cree un grupo de seguridad que utilizará para sus clústeres. En Inbound Rules (Reglas de entrada), seleccione Edit Inbound Rules (Editar reglas de entrada) y, a continuación, seleccione Add Rule (Agregar regla). Este grupo de seguridad permitirá el acceso a los miembros de otro grupo de seguridad.
4. En Type (Tipo), elija Custom TCP Rule (Personalizar regla de TCP).

   a. En Port Range (Rango de puerto), especifique el puerto que utilizó al crear su clúster.

      El puerto predeterminado para los clústeres de MemoryDB es 6379.

   b. En el cuadro Source (Fuente), comience a escribir el ID del grupo de seguridad. Desde la lista, seleccione el grupo de seguridad que utilizará para sus instancias de Amazon EC2.
5. Cuando haya terminado, elija Save (Guardar).

Una vez que haya habilitado el acceso, se encontrará listo para conectarse al clúster, como se describe en la siguiente sección.

Para obtener información sobre cómo acceder a su clúster de MemoryDB desde una Amazon VPC diferente, una AWS región diferente o incluso su red corporativa, consulte lo siguiente:

- Patrones de acceso para obtener acceso a un clúster de MemoryDB en una Amazon VPC
- Acceso a los recursos de MemoryDB desde fuera de AWS

Paso 3: conectar al clúster

Antes de continuar, realice el Paso 2: autorizar el acceso al clúster.

En esta sección se da por sentado que ha creado una instancia de Amazon EC2 y que se puede conectar a ella. Para obtener instrucciones al respecto, consulte la Guía de introducción a Amazon EC2.

Una instancia de Amazon EC2 se puede conectar a un clúster solo si lo ha autorizado.

Encontrar el punto de conexión de un clúster

Una vez que el clúster se encuentra en estado disponible y ha autorizado el acceso a él, puede iniciar sesión en una instancia de Amazon EC2 y conectarse al clúster. Para ello, primero debe determinar el punto de conexión.

Para buscar puntos de conexión, consulte el siguiente enlace:

- Búsqueda del punto de conexión para un clúster de MemoryDB (AWS Management Console)
- Búsqueda del punto de conexión para un clúster de MemoryDB (CLI de AWS)
- Búsqueda del punto de conexión para un clúster de MemoryDB (API de MemoryDB)

Conectarse a un clúster de MemoryDB (Linux)

Ahora que tiene el punto de conexión que necesita, puede iniciar sesión en una instancia EC2 y conectarse al clúster. En el siguiente ejemplo, utilice la utilidad de la CLI para conectarse a un clúster mediante Ubuntu 22. La última versión de la CLI también admite SSL/TLS a fin de conectar clústeres habilitados para cifrado y autenticación.

Conexión a los nodos de MemoryDB mediante redis-cli

Para obtener acceso a los datos desde nodos de MemoryDB, se utilizan clientes que utilizan la capa de conexión segura (SSL). También puede utilizar redis-cli con TLS/SSL en Amazon Linux y Amazon Linux 2.

Para utilizar redis-cli para conectarse a un clúster de MemoryDB en Amazon Linux 2 o Amazon Linux

1. Descargue y compile la utilidad redis-cli. Esta utilidad se incluye en la distribución de software de Redis.

2. En la línea de comandos de su instancia EC2, escriba los comandos correspondientes a la versión de Linux que esté utilizando.

   Amazon Linux 2023

   Si utiliza Amazon Linux 2023, introduzca lo siguiente:

   ```
   sudo yum install redis6 -y
   ```

   A continuación, escriba el siguiente comando y sustituya el punto final del clúster y el puerto por lo que se muestra en este ejemplo.

   ```
   redis-cli -h Primary or Configuration Endpoint --tls -p 6379
   ```

   Para obtener más información sobre cómo encontrar el punto de conexión, consulte Encontrar los puntos de conexión de los nodos.

   Amazon Linux 2

   Si utiliza Amazon Linux 2, introduzca lo siguiente:

   ```
   sudo yum -y install openssl-devel gcc
   wget http://download.redis.io/redis-stable.tar.gz
   tar xvzf redis-stable.tar.gz
   cd redis-stable
   make distclean
   make redis-cli BUILD_TLS=yes
   sudo install -m 755 src/redis-cli /usr/local/bin/
   ```

   Amazon Linux

   Si utiliza Amazon Linux, introduzca lo siguiente:

   ```
   sudo yum install gcc jemalloc-devel openssl-devel tcl tcl-devel clang wget
   wget http://download.redis.io/redis-stable.tar.gz
   tar xvzf redis-stable.tar.gz
   cd redis-stable
   make redis-cli CC=clang BUILD_TLS=yes
   sudo install -m 755 src/redis-cli /usr/local/bin/
   ```

En Amazon Linux, es posible que también deba ejecutar los siguientes pasos adicionales:

```
sudo yum install clang
CC=clang make
sudo make install
```

3. Tras descargar e instalar la utilidad redis-cli, se recomienda ejecutar el comando opcional. `make-test`
4. Para conectarse a un clúster con el cifrado y la autenticación habilitados, introduzca este comando:

```
redis-cli -h Primary or Configuration Endpoint --tls -a 'your-password' -p 6379
```

> **Note**
>
> Si instala redis6 en Amazon Linux 2023, ahora puede usar `redis6-cli` el comando en lugar de: `redis-cli`
>
> ```
> redis6-cli -h Primary or Configuration Endpoint --tls -p 6379
> ```

Paso 4: eliminar un clúster

Siempre que un clúster tenga el estado available, se cobrará por él, independientemente de si lo esté usando de forma activa o no. Para que dejen de devengarse cargos, elimine el clúster.

> **Warning**
>
> Cuando se elimina un clúster de MemoryDB, las instantáneas manuales se conservan. También puede crear una instantánea final antes de eliminar el clúster. Por el contrario, las instantáneas automáticas no se conservan. Para obtener más información, consulte Instantánea y restauración .

Usando el AWS Management Console

El siguiente procedimiento elimina un único clúster de su implementación. Para eliminar varios clústeres, repita el procedimiento por cada clúster que desee eliminar. No es necesario esperar a un clúster para terminar de eliminarlo antes de empezar el procedimiento para eliminar otro clúster.

Para eliminar un clúster

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. Para elegir el clúster que desea eliminar, elija el botón de opción situado junto al nombre del clúster en la lista de clústeres. En este caso, el nombre del clúster que creó en Paso 1: creación de un clúster.
3. En Actions (Acciones), seleccione Delete (Eliminar).
4. Primero, elija si desea crear una instantánea del clúster antes de eliminarlo y, a continuación, escriba `delete` en el cuadro de confirmación y seleccione Eliminar para eliminar el clúster, o bien elija Cancelar para conservar el clúster.

   Si elige Delete, el estado del clúster cambia a deleting.

En cuanto el clúster desaparezca de la lista de clústeres, dejará de incurrir en gastos.

Usando la AWS CLI

El código siguiente elimina el clúster `my-cluster`. En este caso, sustituya `my-cluster` con el nombre del clúster que creó en Paso 1: creación de un clúster.

```
aws memorydb delete-cluster --cluster-name my-cluster
```

La operación de la CLI `delete-cluster` solo elimina un clúster. Para eliminar varios clústeres, llame a `delete-cluster` por cada clúster que desee eliminar. No es necesario esperar a que se termine de eliminar un clúster antes de eliminar otro.

Para Linux, macOS o Unix:

```
aws memorydb delete-cluster \
    --cluster-name my-cluster \
    --region us-east-1
```

Para Windows:

```
aws memorydb delete-cluster ^
    --cluster-name my-cluster ^
    --region us-east-1
```

Para obtener más información, consulte `delete-cluster`.

Uso de la API de MemoryDB

El código siguiente elimina el clúster `my-cluster`. En este caso, sustituya `my-cluster` con el nombre del clúster que creó en Paso 1: creación de un clúster.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DeleteCluster
    &ClusterName=my-cluster
    &Region=us-east-1
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210802T220302Z
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210802T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210802T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

La operación de API `DeleteCluster` solo elimina un clúster. Para eliminar varios clústeres, llame a `DeleteCluster` por cada clúster que desee eliminar. No es necesario esperar a que se termine de eliminar un clúster antes de eliminar otro.

Para obtener más información, consulte DeleteCluster.

¿Qué tengo que hacer ahora?

Ahora que ha probado el ejercicio de introducción, puede explorar las secciones siguientes para obtener más información acerca de MemoryDB y las herramientas disponibles:

- Empezando con AWS
- Herramientas para Amazon Web Services

- Interfaz de la línea de comandos de AWS
- Referencia de la API de MemoryDB para Redis.

Administración de nodos

Un nodo es el componente básico más pequeño de toda implementación de MemoryDB para Redis. Un nodo pertenece a una partición que pertenece a un clúster. Cada nodo ejecuta la versión del motor que se eligió cuando el clúster se creó o se modificó por última vez. Cada nodo tiene su propio puerto y nombre de servicio de nombres de dominio (DNS). Se admiten varios tipos de nodos de MemoryDB, cada uno de los cuales tiene asociada una cantidad de memoria y unos recursos informáticos diferentes.

Temas

Entre las opciones importantes en relación con los nodos, se encuentran las siguientes:

- Agregar/eliminar nodos de un clúster
- Escalado
- Búsqueda de puntos de conexión

Nodos y particiones de MemoryDB

Una partición es una organización jerárquica de nodos, cada uno de ellos encapsulado en un clúster. Las particiones son compatibles con la reproducción. En una partición, un nodo funciona como nodo principal de lectura/escritura. Todos los demás nodos de la partición funcionan como réplicas de solo lectura del nodo principal. MemoryDB admite varias particiones dentro de un clúster. Esto permite la partición de los datos en un clúster de MemoryDB.

MemoryDB admite la replicación mediante particiones. La operación de la API DescribeClustersenumera los fragmentos con los nodos miembros, los nombres de los nodos, los puntos finales y también otra información.

Después de crear un clúster de MemoryDB, se puede modificar (escalarse o reducirse horizontalmente). Para obtener más información, consulte Escalado y Sustitución de nodos.

Cuando cree un clúster nuevo, puede inicializarlo con datos del clúster anterior para que no comience vacío. Hacerlo puede resultar útil si necesita cambiar el tipo de nodo, la versión del motor o migrar desde Amazon ElastiCache for Redis. Para obtener más información, consulte Toma de instantáneas manuales y Restauración a partir de una instantánea.

Tipos de nodos compatibles

MemoryDB admite los siguientes tipos de nodos.

Optimizada para memoria

Tipo de instancia	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)	Multiplexación de E/S mejorada (Redis 7.0.4+)	Versión mínima del motor
db.r7g.large	0.937	12,5	No	6.2
db.r7g.xlarge	1.876	12,5	No	6.2
db.r7g.2xlarge	3.75	15	Sí	6.2
db.r7g.4xlarge	7.5	15	Sí	6.2
db.r7g.8xlarge	15	N/A	Sí	6.2
db.r7g.12xlarge	22,5	N/A	Sí	6.2
db.r7g.16xlarge	30	N/A	Sí	6.2
db.r6g.large	0.75	10.0	No	6.2
db.r6g.xlarge	1,25	10.0	No	6.2
db.r6g.2xlarge	2,5	10.0	Sí	6.2
db.r6g.4xlarge	5.0	10.0	Sí	6.2
db.r6g.8xlarge	12	N/A	Sí	6.2
db.r6g.12xlarge	20	N/A	Sí	6.2
db.r6g.16xlarge	25	N/A	Sí	6.2

Memoria optimizada con la organización de datos en niveles

Tipo de instancia	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)	Multiplexación de E/S mejorada (Redis 7.0.4+)	Versión mínima del motor
db.r6gd.xlarge	1.25	10	No	6.2
db.r6gd.2xlarge	2,5	10	No	6.2
db.r6gd.4xlarge	5.0	10	No	6.2
db.r6gd.8xlarge	12	N/A	No	6.2

Nodos de uso general

Tipo de instancia	Banda ancha de base (Gbps)	Banda ancha con ráfagas (Gbps)	Multiplexación de E/S mejorada (Redis 7.0.4+)	Versión mínima del motor
db.t4g.small	0.128	5.0	No	6.2
db.t4g.medium	0,256	5.0	No	6.2

Para conocer AWS la disponibilidad regional, consulte los precios de MemoryDB for Redis

Todos los tipos de nodos se crean en una nube privada virtual (VPC).

Nodos reservados de MemoryDB

Los nodos reservados ofrecen un descuento importante en comparación con los precios de los nodos bajo demanda. Los nodos reservados no son nodos físicos, sino más bien un descuento de facturación que se aplica al uso de nodos bajo demanda en su cuenta. Los descuentos para los nodos reservados dependen del tipo de nodo y AWS de la región.

El proceso general para trabajar con nodos reservados es el siguiente:

- Revisar información acerca de las ofertas de nodos reservados disponibles
- Adquiera una oferta de nodo reservado mediante el AWS Management Console, AWS Command Line Interface o el SDK
- Revise la información sobre sus nodos reservados existentes

Temas

Información general sobre los nodos reservados

Cuando adquiere un nodo reservado de MemoryDB, adquiere un compromiso para obtener una tarifa con descuento en un tipo de nodo específico durante el periodo de duración del nodo reservado. Para usar un nodo reservado de MemoryDB, debe crear un nodo nuevo, tal como haría para un nodo bajo demanda. El nodo nuevo que cree deberá tener exactamente las mismas especificaciones que el nodo reservado. Si las especificaciones del nuevo nodo coinciden con un nodo reservado existente de su cuenta, se facturará con la tarifa con descuento ofrecida para el nodo reservado. De lo contrario, el nodo se factura con una tarifa bajo demanda. Puede usar la API AWS Management Console, la o la AWS CLI API de MemoryDB para enumerar y comprar las ofertas de nodos reservados disponibles.

MemoryDB ofrece nodos reservados para los nodos R7g, R6g y R6gd optimizados para la memoria (con estratificación de datos). Para obtener información acerca de los precios, consulte Precios de MemoryDB para Redis.

Tipos de ofertas

Los nodos reservados están disponibles en tres variedades: sin pago inicial, pago inicial parcial y pago inicial total, lo cual le permite optimizar sus costos de MemoryDB para Redis en función del uso previsto.

Sin pago inicial: esta opción proporciona acceso a un nodo reservado sin que haya que hacer un pago inicial. Su nodo reservado sin pago inicial le cobra una tarifa por hora con descuento por cada hora dentro del plazo, independientemente del uso. No es necesario realizar ningún pago inicial.

Pago inicial parcial: esta opción exige que parte del nodo reservado se pague por adelantado. Las horas restantes del término se cobran a una tarifa por hora con descuento, independientemente de la utilización que haga.

Pago inicial total: se realiza un pago total al comienzo del plazo, y no se aplicará ningún otro costo el resto del plazo, independientemente del número de horas de uso.

Los tres tipos de ofertas están disponibles en plazos de un año y de tres años.

Tamaño de los nodos reservados con flexibilidad

Al comprar un nodo reservado, una de las cosas que especifica es el tipo de nodo, por ejemplo, db.r6g.xlarge. Para obtener más información sobre los tipos de nodos, consulte Precios de MemoryDB para Redis.

Si tiene un nodo y debe escalarlo para aumentar su capacidad, el nodo reservado se aplica automáticamente al nodo escalado. Es decir, los nodos reservados se aplican automáticamente al uso de cualquier tamaño en la misma familia de nodos. Los nodos reservados de tamaño flexible están disponibles para los nodos de la misma región. AWS Los nodos reservados con flexibilidad de tamaño solo se pueden reducir horizontalmente en sus familias de nodos. Por ejemplo, un nodo reservado para db.r6g.xlarge puede aplicarse a db.r6g.2xlarge, pero no a db.r6gd.large, porque db.r6g y db.r6gd son familias de nodos diferentes.

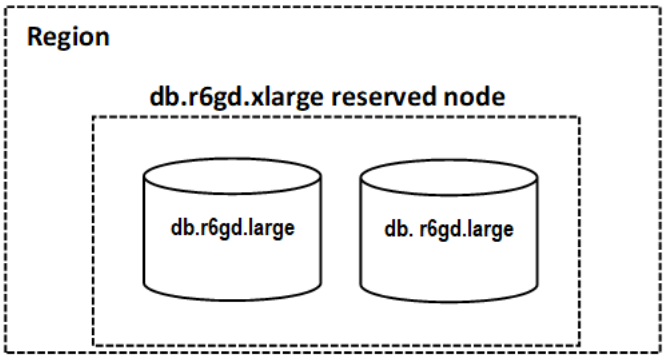
La flexibilidad de tamaño significa que puede moverse libremente entre configuraciones dentro de la misma familia de nodos. Por ejemplo, puede pasar de un nodo reservado r6g.xlarge (8 unidades normalizadas) a dos nodos reservados r6g.large (8 unidades normalizadas) (2*4 = 8 unidades normalizadas) en la misma región sin coste adicional. AWS

Puede comparar el uso de diferentes tamaños de nodos reservados utilizando unidades normalizadas. Por ejemplo, una unidad de uso en dos nodos db.r6g.4xlarge equivale a 16 unidades

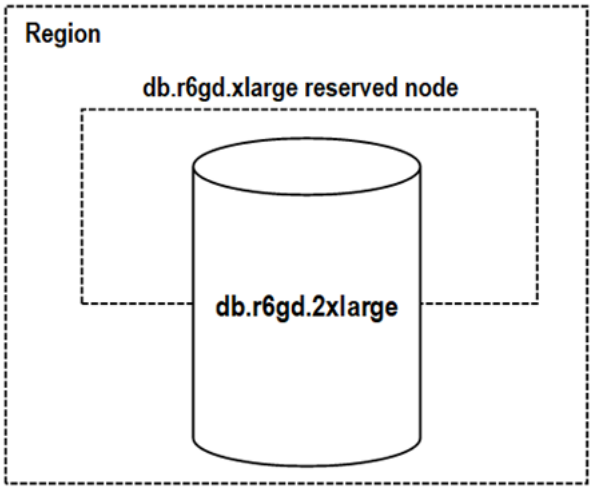
normalizadas de uso en uno db.r6g.large. En la tabla siguiente se muestra el número de unidades normalizadas por cada tamaño de nodo:

Tamaño del nodo	Unidades normalizadas
small	1
medium	2
large	4
xlarge	8
2xlarge	16
4xlarge	32
6xlarge	48
8xlarge	64
10xlarge	80
12xlarge	96
16xlarge	128

Por ejemplo, compra un nodo reservado db.r6gd.xlarge y tiene dos nodos reservados db.r6gd.large en ejecución en su cuenta en la misma región. AWS En este caso, el beneficio de facturación se aplica en su totalidad a los dos nodos.



Como alternativa, si tiene una instancia de db.r6gd.2xlarge ejecutándose en su cuenta de la misma región, el beneficio de facturación se aplica al 50 por ciento del uso del nodo reservado. AWS



Eliminación de un nodo reservado

Los términos de un nodo reservado implican un compromiso de un año o de tres años. No se puede cancelar un nodo reservado. Sin embargo, puede eliminar un nodo que tenga un descuento de nodo reservado. El proceso para eliminar un nodo con un descuento de nodo reservado es el mismo que para cualquier otro nodo.

Si elimina un nodo con un descuento de nodo reservado, puede lanzar otro nodo con especificaciones compatibles. En este caso, sigue disfrutando de la tarifa de descuento mientras dure la reserva (de uno o tres años).

Trabajar con los nodos reservados

Puede usar la API AWS Management Console, la y MemoryDB para AWS Command Line Interface trabajar con nodos reservados.

Consola

Para obtener precios e información sobre ofertas de nodos reservados disponibles

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación, seleccione Nodos reservados.
3. Seleccione Adquirir nodos reservados.
4. En Tipo de nodo, elija el tipo de nodo que desea implementar.

5. En Cantidad, elija la cantidad de nodos que desea implementar.
6. En Plazo, elija el tiempo durante el cual desea que se reserve el nodo de base de datos.
7. En Offering type (Tipo de oferta), elija el tipo de oferta.

Después de realizar estas selecciones, podrá ver la información sobre los precios en Resumen de reserva.

> **Important**
>
> Elija Cancelar para evitar comprar estos nodos reservados e incurrir en cualquier gasto.

Después de recibir la información sobre las ofertas disponibles de nodos reservados, podrá utilizar dicha información para adquirir una oferta, tal como se explica a continuación:

Para adquirir un nodo reservado

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación, seleccione Nodos reservados.
3. Seleccione Adquirir nodos reservados.
4. En Tipo de nodo, elija el tipo de nodo que desea implementar.
5. En Cantidad, elija la cantidad de nodos que desea implementar.
6. En Plazo, elija el tiempo durante el cual desea que se reserve el nodo de base de datos.
7. En Offering type (Tipo de oferta), elija el tipo de oferta.
8. (Opcional) Puede asignar su propio identificador a los nodos reservados que adquiera para poder realizar un seguimiento de ellos. En ID de reserva, escriba un identificador para el nodo reservado.

   Después de realizar estas selecciones, podrá ver la información sobre los precios en Resumen de reserva.
9. Seleccione Adquirir nodos reservados.
10. Los nodos reservados se compran y, a continuación, se muestran en la lista de Nodos reservados.

Para obtener información sobre los nodos reservados para su cuenta AWS

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación, seleccione Nodos reservados.
3. Aparecerán los nodos reservados de la cuenta. Para ver información detallada sobre un nodo reservado en concreto, elija dicho nodo en la lista. Entonces, podrá ver información detallada sobre ese nodo en los detalles.

AWS Command Line Interface

En el siguiente ejemplo de `describe-reserved-nodes-offerings`, se muestran los detalles de las ofertas de nodos reservados.

```
aws memorydb describe-reserved-nodes-offerings
```

Esto debería obtener una salida similar a la siguiente:

```
{
    "ReservedNodesOfferings": [
        {
            "ReservedNodesOfferingId": "0193cc9d-7037-4d49-b332-xxxxxxxxxxxx",
            "NodeType": "db.xxx.large",
            "Duration": 94608000,
            "FixedPrice": $xxx.xx,
            "OfferingType": "Partial Upfront",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": $xx.xx,
                    "RecurringChargeFrequency": "Hourly"
                }
            ]
        }
    ]
}
```

También puede pasar los siguientes parámetros para limitar el alcance de lo que se devuelve:

- `--reserved-nodes-offering-id` – El ID de la oferta que desea adquirir.
- `--node-type`: valor del filtro del tipo de nodo. Utilice este parámetro para mostrar solo las reservas que coincidan con el tipo de nodo especificado.
- `--duration`: valor del filtro de duración, especificado en años o segundos. Utilice este parámetro para mostrar solo las reservas de esta duración.
- `--offering-type`: utilice este parámetro para mostrar solo las ofertas disponibles que coincidan con el tipo de oferta especificado.

Después de recibir la información sobre las ofertas disponibles de nodos reservados, podrá utilizar dicha información para adquirir una oferta.

En el siguiente ejemplo de `purchase-reserved-nodes-offering`, se compran nuevos nodos reservados

Para Linux, macOS o Unix:

```
aws memorydb purchase-reserved-nodes-offering \

    --reserved-nodes-offering-id 0193cc9d-7037-4d49-b332-d5e984f1d8ca \
    --reservation-id reservation \
    --node-count 2
```

Para Windows:

```
aws memorydb purchase-reserved-nodes-offering ^
    --reserved-nodes-offering-id 0193cc9d-7037-4d49-b332-d5e984f1d8ca ^
    --reservation-id MyReservation
```

- `--reserved-nodes-offering-id` representa el nombre de los nodos reservados que se ofrecen a la venta.
- `--reservation-id` es un identificador especificado por el cliente que permite realizar un seguimiento de la reserva.

> **Note**
>
> El ID de reserva es un identificador único especificado por el cliente que permite realizar un seguimiento de la reserva. Si no se especifica este parámetro, MemoryDB genera automáticamente un identificador para la reserva.

- `--node-count` es el número de nodos que se van a reservar. El valor predeterminado es 1.

Esto debería obtener una salida similar a la siguiente:

```
{
    "ReservedNode": {
        "ReservationId": "reservation",
        "ReservedNodesOfferingId": "0193cc9d-7037-4d49-b332-xxxxxxxxxxxx",
        "NodeType": "db.xxx.large",
        "StartTime": 1671173133.982,
        "Duration": 94608000,
        "FixedPrice": $xxx.xx,
        "NodeCount": 2,
        "OfferingType": "Partial Upfront",
        "State": "payment-pending",
        "RecurringCharges": [
            {
                "RecurringChargeAmount": $xx.xx,
                "RecurringChargeFrequency": "Hourly"
            }
        ],
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxxx:reservednode/reservation"
    }
}
```

Después de adquirir los nodos reservados, podrá obtener información sobre los nodos reservados.

El siguiente ejemplo de `describe-reserved-nodes` devuelve información sobre los nodos reservados para esta cuenta.

```
aws memorydb describe-reserved-nodes
```

Esto debería obtener una salida similar a la siguiente:

```
{
    "ReservedNodes": [
        {
            "ReservationId": "ri-2022-12-16-00-28-40-600",
            "ReservedNodesOfferingId": "0193cc9d-7037-4d49-b332-xxxxxxxxxxxx",
            "NodeType": "db.xxx.large",
            "StartTime": 1671150737.969,
            "Duration": 94608000,
            "FixedPrice": $xxx.xx,
            "NodeCount": 1,
            "OfferingType": "Partial Upfront",
            "State": "active",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": $xx.xx,
                    "RecurringChargeFrequency": "Hourly"
                }
            ],
            "ARN": "arn:aws:memorydb:us-east-1:xxxxxxxx:reservednode/
ri-2022-12-16-00-28-40-600"
        }
    ]
}
```

También puede pasar los siguientes parámetros para limitar el alcance de lo que se devuelve:

- `--reservation-id`: puede asignar su propio identificador a los nodos reservados que adquiera para poder realizar un seguimiento de estos.
- `--reserved-nodes-offering-id`: valor del filtro del identificador de la oferta. Utilice este parámetro para mostrar solo las reservas compradas que coincidan con el identificador de oferta especificado.
- `--node-type`: valor del filtro del tipo de nodo. Utilice este parámetro para mostrar solo las reservas que coincidan con el tipo de nodo especificado.
- `--duration`: valor del filtro de duración, especificado en años o segundos. Utilice este parámetro para mostrar solo las reservas de esta duración.
- `--offering-type`: utilice este parámetro para mostrar solo las ofertas disponibles que coincidan con el tipo de oferta especificado.

API de MemoryDB

Los siguientes ejemplos muestran cómo utilizar la API de consulta de MemoryDB para los nodos reservados:

DescribeReservedNodesOfferings

Devuelve los detalles de las ofertas de nodos reservados.

```
https://memorydb.us-west-2.amazonaws.com/
    ?Action=DescribeReservedNodesOfferings
    &ReservedNodesOfferingId=649fd0c8-xxxx-xxxx-xxxx-06xxxx75e95f
 &"Duration": 94608000,
    &NodeType="db.r6g.large"
    &OfferingType="Partial Upfront"
    &Version=2021-01-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Los siguientes parámetros limitan el alcance de lo que se devuelve:

- `ReservedNodesOfferingId` representa el nombre de los nodos reservados que se ofrecen a la venta.
- `Duration`: valor del filtro de duración, especificado en años o segundos. Utilice este parámetro para mostrar solo las reservas de esta duración.
- `NodeType`: valor del filtro del tipo de nodo. Utilice este parámetro para mostrar solo las ofertas que coincidan con el tipo de nodo especificado.
- `OfferingType`: utilice este parámetro para mostrar solo las ofertas disponibles que coincidan con el tipo de oferta especificado.

Después de recibir la información sobre las ofertas disponibles de nodos reservados, podrá utilizar dicha información para adquirir una oferta.

PurchaseReservedNodesOffering

Permite adquirir una oferta de nodos reservados.

```
https://memorydb.us-west-2.amazonaws.com/
    ?Action=PurchaseReservedCacheNodesOffering
    &ReservedNodesOfferingId=649fd0c8-xxxx-xxxx-xxxx-06xxxx75e95f
    &ReservationID=myreservationID
    &NodeCount=1
    &Version=2021-01-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20141201T220302Z
    &X-Amz-Algorithm
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

- `ReservedNodesOfferingId` representa el nombre de los nodos reservados que se ofrecen a la venta.
- `ReservationID` es un identificador especificado por el cliente que permite realizar un seguimiento de la reserva.

> **Note**
>
> El ID de reserva es un identificador único especificado por el cliente que permite realizar un seguimiento de la reserva. Si no se especifica este parámetro, MemoryDB genera automáticamente un identificador para la reserva.

- `NodeCount` es el número de nodos que se van a reservar. El valor predeterminado es 1.

Después de adquirir los nodos reservados, podrá obtener información sobre los nodos reservados.

DescribeReservedNodes

Devuelve información sobre los nodos reservados para esta cuenta.

```
https://memorydb.us-west-2.amazonaws.com/
 ?Action=DescribeReservedNodes
 &ReservedNodesOfferingId=649fd0c8-xxxx-xxxx-xxxx-06xxxx75e95f
 &ReservationID=myreservationID
 &NodeType="db.r6g.large"
```

```
&Duration=94608000
&OfferingType="Partial Upfront"
&Version=2021-01-01
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

Los siguientes parámetros limitan el alcance de lo que se devuelve:

- `ReservedNodesOfferingId` representa el nombre del nodo reservado.
- `ReservationID`: puede asignar su propio identificador a los nodos reservados que adquiera para poder realizar un seguimiento de estos.
- `NodeType`: valor del filtro del tipo de nodo. Utilice este parámetro para mostrar solo las reservas que coincidan con el tipo de nodo especificado.
- `Duration`: valor del filtro de duración, especificado en años o segundos. Utilice este parámetro para mostrar solo las reservas de esta duración.
- `OfferingType`: utilice este parámetro para mostrar solo las ofertas disponibles que coincidan con el tipo de oferta especificado.

Visualización de la facturación de los nodos reservados

Puede ver la facturación de los nodos reservados en el panel de facturación en la AWS Management Console.

Para ver la facturación de los nodos reservados

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. En el botón de búsqueda de la parte superior de la consola, selecciona Facturación.
3. Selecciona Facturas en la parte izquierda del panel de control.
4. En Cargos por servicio de AWS , expanda MemoryDB.
5. Amplíe la AWS región en la que se encuentran sus nodos reservados, por ejemplo, EE. UU. Este (Norte de Virginia).

Los nodos reservados y sus cargos por hora del mes actual se muestran en Instancias CreateCluster reservadas de Amazon MemoryDB.

Amazon MemoryDB CreateCluster Reserved Instances		
AmazonMemoryDB, db.r6g.large reserved instance applied	81.000 Hrs	
AmazonMemoryDB, db.r6g.4xlarge reserved instance applied	324.000 Hrs	
AmazonMemoryDB, db.r6g.4xlarge reserved instance applied	162.000 Hrs	
USD (hourly fee per AmazonMemoryDB, db.r6g.large instance	1,488.000 Hrs	
USD (hourly fee per AmazonMemoryDB, db.r6gd.2xlarge instance	744.000 Hrs	
USD 0 hourly fee per AmazonMemoryDB, db.r6g.4xlarge instance	744.000 Hrs	
USD (hourly fee per AmazonMemoryDB, db.r6gd.xlarge instance	744.000 Hrs	
USD 1 hourly fee per AmazonMemoryDB, db.r6gd.4xlarge instance	2,976.000 Hrs	

Sustitución de nodos

MemoryDB suele actualizar su flota con parches y actualizaciones, por lo general sin problemas. Sin embargo, cada cierto tiempo tenemos que relanzar los nodos de MemoryDB con el fin de aplicar las actualizaciones obligatorias del sistema operativo en el host subyacente. Estas sustituciones son necesarias para aplicar actualizaciones que refuerzan la seguridad, la fiabilidad y el rendimiento operativo.

Puede optar por administrar personalmente estas sustituciones en cualquier momento antes del periodo programado para la sustitución de nodos. Cuando administre personalmente una sustitución, la instancia recibirá la actualización del sistema operativo cuando vuelva a lanzar el nodo y se cancelará la sustitución de nodos programada. Es posible que reciba alertas que indiquen que va a tener lugar la sustitución de nodos. Si ya ha mitigado manualmente la necesidad de mantenimiento, puede hacer caso omiso de estas alertas.

> **Note**
>
> Los nodos de reemplazo generados automáticamente por MemoryDB para Redis pueden tener direcciones IP diferentes. Usted es responsable de revisar la configuración de la aplicación para asegurarse de que los nodos estén asociados con las direcciones IP apropiadas.

La lista siguiente identifica acciones que puede realizar cuando MemoryDB programa el reemplazo de uno de sus nodos:

Opciones de sustitución de nodos de MemoryDB

- No hacer nada: si no hace nada, MemoryDB reemplaza el nodo según lo programado.

Si el nodo es miembro de un clúster Multi-AZ, MemoryDB proporciona mayor disponibilidad durante la aplicación de parches, las actualizaciones y otras operaciones de sustitución de nodos relacionadas con el mantenimiento.

La sustitución se completa mientras el clúster atiende solicitudes de escritura entrantes.

- Cambie el periodo de mantenimiento: para eventos de mantenimiento programados, recibirá un correo electrónico o un evento de notificación de MemoryDB. En estos casos, si cambia el periodo de mantenimiento antes de la hora de sustitución programada, su nodo se sustituirá en ese momento a la nueva hora. Para obtener más información, consulte Modificación de un clúster de MemoryDB.

  > **Note**
  >
  > La posibilidad de mover el periodo de sustitución para modificarlo solo está disponible cuando la notificación de MemoryDB incluye un periodo de mantenimiento. Si la notificación no incluye un periodo de mantenimiento, no se puede cambiar el periodo de sustitución.

  Por ejemplo, supongamos que es jueves 9 de noviembre a las 15:00 h y el próximo periodo de mantenimiento es el viernes 10 de noviembre a las 17:00 h. A continuación, se exponen tres escenarios con sus resultados:

  - Cambia el periodo de mantenimiento a los viernes a las 16:00, después de la fecha y hora actuales y antes del siguiente periodo de mantenimiento programado. El nodo se sustituye el viernes 10 de noviembre a las 16:00.
  - Cambia el periodo de mantenimiento al sábado a las 16:00, después de la fecha y hora actuales y después del siguiente periodo de mantenimiento programado. El nodo se sustituye el sábado 11 de noviembre a las 16:00.
  - Cambia el periodo de mantenimiento al miércoles a las 16:00 un día anterior de la misma semana que la fecha y hora actuales. El nodo se sustituye el próximo miércoles 15 de noviembre a las 16:00.

  Para obtener instrucciones, consulte Administración del mantenimiento.

Administración de clústeres

La mayoría de las operaciones de MemoryDB se realizan en el clúster. Puede configurar un clúster con un número específico de nodos y un grupo de parámetros que controle las propiedades de cada nodo. Todos los nodos de un clúster están diseñados para ser del mismo tipo y tener los mismos valores de configuración de parámetros y grupo de seguridad.

Cada clúster debe tener un identificador de clúster. El identificador del clúster es un nombre suministrado por el cliente para el clúster. Este identificador especifica un clúster determinado al interactuar con los comandos de la API de MemoryDB y de la AWS CLI. El identificador del clúster debe ser único para ese cliente en una región de AWS.

Los clústeres de MemoryDB se han diseñado para poder obtener acceso a ellos mediante una instancia de Amazon EC2. Solo se puede lanzar el clúster de MemoryDB en una nube privada virtual (VPC) en función del servicio de Amazon VPC, se puede acceder a él desde fuera de AWS. Para obtener más información, consulte Acceso a los recursos de MemoryDB desde fuera de AWS.

Organización de datos en niveles

Los clústeres que utilizan un tipo de nodo de la familia r6gd tienen sus datos en niveles entre la memoria y el almacenamiento SSD local (unidades de estado sólido). La organización de datos en niveles proporciona una nueva opción de precio-rendimiento para las cargas de trabajo de Redis mediante el uso de unidades de estado sólido (SSD) de menor costo en cada nodo de clúster, además de almacenar datos en la memoria. Al igual que en otros tipos de nodos, los datos escritos en los nodos r6gd se almacenan de forma duradera en un registro de transacciones Multi-AZ. La organización de datos en niveles es ideal para cargas de trabajo que acceden regularmente hasta un 20 % de su conjunto de datos general y para aplicaciones que pueden tolerar latencia adicional al acceder a los datos en SSD.

En clústeres con organización de datos en niveles, MemoryDB supervisa la última hora de acceso de cada elemento que almacena. Cuando la memoria disponible (DRAM) se consume por completo, MemoryDB utiliza un algoritmo de menos usados recientemente (LRU) para trasladar automáticamente los elementos a los que se obtiene acceso con poca frecuencia de la memoria a la SSD. Cuando se obtiene acceso posteriormente a los datos de SSD, MemoryDB los mueve de nuevo a la memoria de forma automática y asíncrona antes de procesar la solicitud. Si tiene una carga de trabajo que solo accede a un subconjunto de sus datos regularmente, la organización de datos en niveles es una forma óptima de escalar su capacidad de forma rentable.

Tenga en cuenta que cuando se utiliza la organización de datos en niveles, las propias claves siempre permanecen en la memoria, mientras que la LRU rige la ubicación de los valores en la memoria frente al disco. En general, recomendamos que los tamaños de las claves sean más pequeños que los tamaños de los valores al usar la organización de datos en niveles.

La organización de datos en niveles está diseñada para tener un impacto mínimo en el rendimiento de las cargas de trabajo Por ejemplo, suponiendo valores de cadena de 500 bytes, puede esperar por lo general 450 microsegundos adicionales de latencia para las solicitudes de lectura de datos almacenados en SSD en comparación con las solicitudes de datos de la memoria.

Con el mayor tamaño de nodo de organización de datos en niveles (db.r6gd.8xlarge), puede almacenar hasta ~500 TB en un solo clúster de 500 nodos (250 TB cuando se utiliza 1 réplica de lectura). Para la organización de datos en niveles, MemoryDB reserva el 19 % de la memoria (DRAM) por nodo para usos distintos de los datos. La organización de datos en niveles es compatible con todos los comandos y estructuras de datos de Redis compatibles con MemoryDB. No es necesario cambiar el lado del cliente para usar esta característica.

Temas

Prácticas recomendadas

Recomendamos que siga las siguientes prácticas recomendadas:

- La organización de datos en niveles es ideal para cargas de trabajo que acceden regularmente hasta un 20 % de su conjunto de datos general y para aplicaciones que pueden tolerar latencia adicional al acceder a los datos en SSD.
- Al utilizar la capacidad de SSD disponible en nodos con niveles de datos, recomendamos que el tamaño del valor sea mayor que el tamaño de la clave. El tamaño del valor no puede ser superior a 128 MB; de lo contrario, no se moverá al disco. Cuando se mueven elementos entre DRAM y SSD, las claves siempre permanecerán en la memoria y solo los valores se moverán al nivel de SSD.

Limitaciones

La organización de datos en niveles tiene las siguientes restricciones:

- El tipo de nodo que utilice debe pertenecer a la familia r6gd, que está disponible en las siguientes regiones: `us-east-2`, `us-east-1`, `us-west-2`, `us-west-1`, `eu-west-1`, `eu-west-3`, `eu-central-1`, `ap-northeast-1`, `ap-southeast-1`, `ap-southeast-2`, `ap-south-1`, `ca-central-1` y `sa-east-1`.
- No se puede restaurar una instantánea de un clúster r6gd en otro clúster a menos que también utilice r6gd.
- No se puede exportar una instantánea a Amazon S3 para clústeres de organización de datos en niveles.
- No se admite el guardado sin ramificación.

- No se admite el escalado desde un clúster de organización de datos en niveles (por ejemplo, un clúster que utiliza un tipo de nodo r6gd) a un clúster que no utiliza la organización de datos en niveles (por ejemplo, un clúster que utiliza un tipo de nodo r6g).
- La organización de datos en niveles solo admite las políticas maxmemory `volatile-lru`, `allkeys-lru` y `noeviction`.
- Los artículos de más de 128 MiB no se mueven a SSD.

Precios de organización de datos en niveles

Los nodos R6gd tienen 5 veces más capacidad total (memoria + SSD) y pueden ayudarle a ahorrar más del 60 por ciento de los costos de almacenamiento cuando se ejecutan con la máxima utilización en comparación con los nodos R6g (solo memoria). Para obtener más información, consulte Precios de MemoryDB.

Supervisión

MemoryDB ofrece métricas diseñadas específicamente para monitorear los clústeres de rendimiento que utilizan la organización de datos en niveles. Para monitorear la proporción de elementos en DRAM en comparación con SSD, puede utilizar la métrica `CurrItems` en Métricas de MemoryDB. Puede calcular el porcentaje de la siguiente manera:`(CurrItems with Dimension: Tier = Memory * 100) / (CurrItems with no dimension filter)`. Cuando el porcentaje de elementos en la memoria disminuye por debajo del 5 %, le recomendamos que lo considereEscalado de clústeres de MemoryDB.

Para obtener más información, consulte Métricas para clústeres de MemoryDB que utilizan la organización de datos en niveles en Métricas de MemoryDB.

Uso de la organización de datos en niveles

Uso de la organización de datos en niveles mediante la AWS Management Console

Al crear un clúster, se utiliza la organización de datos en niveles seleccionando un tipo de nodo de la familia r6gd, como db.r6gd.xlarge. La selección de ese tipo de nodo habilita automáticamente la organización de datos en niveles.

Para obtener más información sobre la creación de clústeres, consulte Paso 1: creación de un clúster.

Activación de la organización de datos en niveles mediante la AWS CLI

Al crear un clúster utilizando la AWS CLI, puede utilizar la organización de datos en niveles seleccionando un tipo de nodo de la familia r6gd, como db.r6gd.xlarge y configurando el parámetro `--data-tiering`.

No puede excluirse de la organización de datos en niveles al seleccionar un tipo de nodo de la familia r6gd. Si configura el parámetro `--no-data-tiering`, la operación no se llevará a cabo correctamente.

Para Linux, macOS o Unix:

```
aws memorydb create-cluster \
   --cluster-name my-cluster \
   --node-type db.r6gd.xlarge \
   --acl-name my-acl \
   --subnet-group my-sg \
   --data-tiering
```

Para Windows:

```
aws memorydb create-cluster ^
   --cluster-name my-cluster ^
   --node-type db.r6gd.xlarge ^
   --acl-name my-acl ^
   --subnet-group my-sg
   --data-tiering
```

Después de ejecutar esta operación, verá una respuesta parecida a la siguiente:

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "creating",
        "NumberOfShards": 1,
        "AvailabilityMode": "MultiAZ",
        "ClusterEndpoint": {
            "Port": 6379
        },
        "NodeType": "db.r6gd.xlarge",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
```

```
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxxxxxxxxx:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "ACLName": "my-acl",
        "DataTiering":"true",
        "AutoMinorVersionUpgrade": true
    }
}
```

Restauración de datos desde una instantánea en clústeres con la organización de datos en niveles habilitada

Puede restaurar una instantánea en un nuevo clúster con la organización de datos en niveles habilitada mediante la consola, la CLI de AWS o la API de MemoryDB. Cuando crea un clúster mediante tipos de nodos de la familia r6gd, se habilita la organización de datos en niveles.

Restauración de datos desde una instantánea en clústeres con la organización de datos en niveles habilitada (consola)

Para restaurar una instantánea a un nuevo clúster con la organización de datos en niveles habilitada (consola), siga los pasos que se indican en Restauración a partir de una instantánea (consola)

Tenga en cuenta que para habilitar la organización de datos en niveles, debe seleccionar un tipo de nodo de la familia r6gd.

Restauración de datos desde una instantánea en clústeres con la organización de datos en niveles habilitada (CLI de AWS)

Al crear un clúster utilizando la AWS CLI, la organización de datos en niveles se utiliza de forma predeterminada seleccionando un tipo de nodo de la familia r6gd, como db.r6gd.xlarge y configurando el parámetro `--data-tiering`.

No puede excluirse de la organización de datos en niveles al seleccionar un tipo de nodo de la familia r6gd. Si configura el parámetro `--no-data-tiering`, la operación no se llevará a cabo correctamente.

Para Linux, macOS o Unix:

```
aws memorydb create-cluster \
   --cluster-name my-cluster \
   --node-type db.r6gd.xlarge \
   --acl-name my-acl \
   --subnet-group my-sg \
   --data-tiering \
   --snapshot-name my-snapshot
```

Para Linux, macOS o Unix:

```
aws memorydb create-cluster ^
   --cluster-name my-cluster ^
   --node-type db.r6gd.xlarge ^
   --acl-name my-acl ^
   --subnet-group my-sg ^
   --data-tiering ^
   --snapshot-name my-snapshot
```

Después de ejecutar esta operación, verá una respuesta parecida a la siguiente:

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "creating",
        "NumberOfShards": 1,
        "AvailabilityMode": "MultiAZ",
        "ClusterEndpoint": {
            "Port": 6379
        },
        "NodeType": "db.r6gd.xlarge",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxxxxxxxxx:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
```

```
        "ACLName": "my-acl",
        "DataTiering": "true"
}
```

Preparación de un clúster

A continuación, puede encontrar instrucciones acerca de cómo crear un clúster a través de la consola de MemoryDB, la AWS CLI o la API de MemoryDB.

Siempre que cree un clúster, es conveniente realizar algunos preparativos para que no sea necesario actualizar o efectuar cambios de inmediato.

Temas

- Determinación de los requisitos

Determinación de los requisitos

Preparación

Conocer las respuestas a las siguientes preguntas ayuda a que la creación del clúster sea más fluida:

- Asegúrese de crear un grupo de subredes en la misma VPC antes de comenzar a crear un clúster. También puede utilizar el grupo de subredes predeterminado proporcionado. Para obtener más información, consulte Subredes y grupos de subredes.

  MemoryDB se ha diseñado para el acceso desde AWS mediante Amazon EC2. Sin embargo, si se lanza en una VPC basada en Amazon VPC, se puede proporcionar acceso desde fuera de AWS. Para obtener más información, consulte Acceso a los recursos de MemoryDB desde fuera de AWS.
- ¿Necesita personalizar los valores de algún parámetro?

  Si lo hace, cree un grupo de parámetros personalizado. Para obtener más información, consulte Creación de un grupo de parámetros.
- ¿Necesita crear un grupo de seguridad de VPC?

  Para obtener más información, consulte Seguridad en la VPC.
- ¿Cómo pretende implementar la tolerancia a errores?

Para obtener más información, consulte Mitigación de errores.

Temas

- Requisitos de procesador y memoria
- Configuración de los clústeres de MemoryDB
- Multiplexación de E/S mejorada
- Requisitos de escalado
- Requisitos de acceso
- Regiones y zonas de disponibilidad

Requisitos de procesador y memoria

El componente básico de MemoryDB para Redis es el nodo. Los nodos se configuran en particiones para formar clústeres. A la hora de determinar el tipo de nodo que desea utilizar para el clúster, tenga en cuenta la configuración del nodo del clúster y la cantidad de datos que tiene que almacenar.

Configuración de los clústeres de MemoryDB

Los clústeres de MemoryDB se componen de 1 a 500 particiones. En un clúster de MemoryDB, los datos están particionados en las distintas particiones del clúster. Su aplicación se conecta con un clúster de MemoryDB mediante una dirección de red denominada punto de conexión. Además de los puntos de conexión del nodo, el clúster de MemoryDB tiene un punto de conexión denominado punto de conexión de clúster. Su aplicación puede usar este punto de conexión para leer o escribir en el clúster y no tener que determinar de qué nodo efectuar la lectura o hasta cuál escribir en MemoryDB.

Multiplexación de E/S mejorada

Si ejecuta la versión 7.0 o superior de Redis, obtendrá una aceleración adicional con la multiplexación de E/S mejorada, en la que cada subproceso de E/S de red dedicado canaliza los comandos de varios clientes al motor de Redis, aprovechando la capacidad de Redis de procesar comandos en lotes de manera eficiente. Para obtener más información, consulte Rendimiento ultrarrápido ythe section called “Tipos de nodos compatibles”.

Requisitos de escalado

Todos los clústeres se pueden ampliar a un tipo de nodo más grande. Al escalar verticalmente un clúster de MemoryDB, puede hacerlo en línea para que el clúster siga disponible o puede crear un nuevo clúster a partir de una instantánea y evitar que el nuevo clúster comience vacío.

Para obtener más información, consulte la sección Escalado de esta guía.

Requisitos de acceso

Por diseño, el acceso a los clústeres de MemoryDB se realiza desde instancias de Amazon EC2. El acceso de red a un clúster de MemoryDB se encuentra limitado a la cuenta que creó el clúster. Por lo tanto, antes de poder obtener acceso a un clúster desde una instancia de Amazon EC2, debe autorizar a dicha instancia el acceso al clúster. Para obtener instrucciones detalladas, consulte Paso 2: autorizar el acceso al clúster en esta guía.

Regiones y zonas de disponibilidad

Al situar sus clústeres de MemoryDB en una región de AWS cercana a su aplicación, puede reducir la latencia. Si el clúster tiene varios nodos, ubicar los nodos en distintas zonas de disponibilidad puede reducir el impacto de los errores en el clúster.

Para obtener más información, consulte los siguientes temas:

- Elección de regiones y zonas de disponibilidad
- Mitigación de errores

Creación de un clúster

MemoryDB para Redis ofrece tres formas de crear un clúster. Para obtener más información, consulte Paso 1: creación de un clúster.

Visualización de los detalles de un clúster

Puede consultar información detallada acerca de uno o varios clústeres mediante la consola de MemoryDB, la AWS CLI o la API de MemoryDB.

Visualización de los detalles de un clúster de MemoryDB (consola)

El siguiente procedimiento detalla cómo consultar los detalles de un clúster de MemoryDB utilizando la consola de MemoryDB.

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. Para ver los detalles de un clúster, elija el botón de opción situado a la izquierda del nombre del clúster y, a continuación, elija Ver detalles. También puede hacer clic directamente en el clúster para ver la página de detalles del clúster.

   La página de detalles del clúster muestra los detalles sobre el clúster, incluido el punto de conexión del clúster. Puede ver más detalles en las múltiples pestañas disponibles en la página de detalles del clúster.
3. Elija la pestaña Particiones y nodos para elegir una lista de las particiones del clúster y el número de nodos en cada partición.
4. Para ver información específica sobre un nodo, expanda la partición en la siguiente tabla. Como alternativa, también puede buscar la partición mediante el cuadro de búsqueda.

   Al hacerlo, se muestra información sobre cada nodo, incluida su zona de disponibilidad, los espacios de claves y los espacios de teclas y su estado.
5. Seleccione la pestaña Métricas para supervisar sus procesos respectivos, como la utilización de la CPU y la utilización de la CPU del motor. Para obtener más información, consulte Métricas de MemoryDB.
6. Seleccione la pestaña Red y seguridad para ver los detalles del grupo de subredes y los grupos de seguridad.

   a. En el grupo de subredes, puede ver el nombre del grupo de subredes, un enlace a la VPC a la que pertenece la subred y el nombre de recurso de Amazon (ARN) del grupo de subredes.

   b. En Grupos de seguridad, puede ver el ID, el nombre y la descripción del grupo de seguridad.

7. Seleccione la pestaña Mantenimiento e instantáneas para ver los detalles de la configuración de las instantáneas.

   a. En Instantánea, puede ver si las instantáneas automatizadas están habilitadas, el periodo de retención de las instantáneas y el periodo de instantáneas.

   b. En Instantáneas, verá una lista de todas las instantáneas de este clúster, con el nombre de la instantánea, el tamaño, la cantidad de particiones y el estado.

   Para obtener más información, consulte Instantánea y restauración .
8. Seleccione las pestañas Mantenimiento e instantáneas para ver los detalles del periodo de mantenimiento, junto con las actualizaciones pendientes de ACL, refragmentación o servicio. Para obtener más información, consulte Administración del mantenimiento.
9. Seleccione la pestaña Actualizaciones de servicio para ver los detalles de cualquier actualización de servicio que se aplique a este clúster. Para obtener más información, consulte Actualizaciones de los servicios de MemoryDB para Redis.
10. Seleccione la pestaña Etiquetas para ver los detalles de cualquier etiqueta de asignación de recursos o costos que esté asociada a este clúster. Para obtener más información, consulte Etiquetado de instantáneas.

Visualización de los detalles de un clúster (CLI de AWS)

Puede ver los detalles de un clúster utilizando el comando AWS CLI de la `describe-clusters`. Si el parámetro `--cluster-name` se omite, se devolverán los detalles de varios clústeres, hasta `--max-results`. Si se incluye el parámetro `--cluster-name`, se devolverán detalles del clúster especificado. Puede limitar el número de registros que devuelve con el parámetro `--max-results`.

El siguiente código enumera los detalles de `my-cluster`.

```
aws memorydb describe-clusters --cluster-name my-cluster
```

El siguiente código enumera los detalles de hasta 25 clústeres.

```
aws memorydb describe-clusters --max-results 25
```

Example

Para Linux, macOS o Unix:

```
aws memorydb describe-clusters \
    --cluster-name my-cluster \
    --show-shard-details
```

Para Windows:

```
aws memorydb describe-clusters ^
    --cluster-name my-cluster ^
    --show-shard-details
```

La siguiente salida JSON muestra la respuesta:

```
{
    "Clusters": [
        {
            "Name": "my-cluster",
            "Description": "my cluster",
            "Status": "available",
            "NumberOfShards": 1,
            "Shards": [
                {
                    "Name": "0001",
                    "Status": "available",
                    "Slots": "0-16383",
                    "Nodes": [
                        {
                            "Name": "my-cluster-0001-001",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": 1629230643.961,
                            "Endpoint": {
                                "Address": "my-cluster-0001-001.my-
cluster.abcdef.memorydb.us-east-1.amazonaws.com",
                                "Port": 6379
                            }
                        },
                        {
                            "Name": "my-cluster-0001-002",
                            "Status": "available",
                            "CreateTime": 1629230644.025,
                            "Endpoint": {
```

```
                                "Address": "my-cluster-0001-002.my-
cluster.abcdef.memorydb.us-east-1.amazonaws.com",
                                "Port": 6379
                            }
                        }
                    ],
                    "NumberOfNodes": 2
                }
            ],
            "ClusterEndpoint": {
                "Address": "clustercfg.my-cluster.abcdef.memorydb.us-
east-1.amazonaws.com",
                "Port": 6379
            },
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "ParameterGroupName": "default.memorydb-redis6",
            "ParameterGroupStatus": "in-sync",
            "SubnetGroupName": "default",
            "TLSEnabled": true,
            "ARN": "arn:aws:memorydb:us-east-1:000000000:cluster/my-cluster",
            "SnapshotRetentionLimit": 0,
            "MaintenanceWindow": "sat:06:30-sat:07:30",
            "SnapshotWindow": "04:00-05:00",
            "ACLName": "open-access",
            "DataTiering": "false",
            "AutoMinorVersionUpgrade": true,
        }
```

Para obtener más información, consulte el tema de la AWS CLI para MemoryDB `describe-clusters`.

Visualización de detalles de un clúster (API de MemoryDB)

Puede ver los detalles de un clúster utilizando la acción `DescribeClusters` de la API de MemoryDB. Si se incluye el parámetro `ClusterName`, se devolverán detalles del clúster especificado. Si el parámetro `ClusterName` se omite, se devolverán los detalles de hasta `MaxResults` clústeres (el valor predeterminado es 100). El valor de `MaxResults` no puede ser inferior a 20 ni superior a 100.

El siguiente código enumera los detalles de `my-cluster`.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DescribeClusters
   &ClusterName=my-cluster
   &Version=2021-01-01
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20210802T192317Z
   &X-Amz-Credential=<credential>
```

El siguiente código enumera los detalles de hasta 25 clústeres.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DescribeClusters
   &MaxResults=25
   &Version=2021-02-02
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20210802T192317Z
   &X-Amz-Credential=<credential>
```

Para obtener más información, consulte el tema de referencia de la API de MemoryDB `DescribeClusters`.

Modificación de un clúster de MemoryDB

Además de agregar o quitar nodos de un clúster, puede que haya veces en las que necesite realizar otros cambios en un clúster existente, como, por ejemplo, agregar un grupo de seguridad o cambiar el periodo de mantenimiento o un grupo de parámetros.

Recomendamos que el periodo de mantenimiento corresponda al momento de mínimo uso. Esto puede requerir alguna modificación de vez en cuando.

Cuando cambia los parámetros de un clúster, el cambio se aplica al clúster inmediatamente. Esto es cierto tanto si se modifica el propio grupo de parámetros del clúster como si se modifica el valor de un parámetro del grupo.

También puede actualizar la versión del motor de sus clústeres. Por ejemplo, puede seleccionar una nueva versión secundaria del motor y MemoryDB empezará a actualizar su clúster inmediatamente.

Uso de la AWS Management Console

Pasos para modificar un clúster

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. En la lista de la esquina superior derecha, elija la región de AWS en la que se encuentra el clúster que desea modificar.
3. En el panel de navegación de la izquierda, vaya a Clústeres. En el detalle de clústeres, seleccione el clúster con el botón de opción y vaya a Acciones y, a continuación, a Modificar.
4. Aparece la página Modificar.
5. En la ventana Modificar, haga las modificaciones que desee. Las opciones son:

   - Descripción
   - Grupos de subredes
   - Grupos de seguridad de VPC.
   - Tipo de nodo

   > **Note**
   >
   > Si el clúster utiliza un tipo de nodo de la familia r6gd, solo puede elegir un tamaño de nodo diferente dentro de esa familia. Si elige un tipo de nodo de la familia r6gd,

la organización de datos en niveles se activará automáticamente. Para obtener más información, consulte Organización de datos en niveles.

- Compatibilidad de versiones de Redis
- Habilitar instantáneas automáticas
- Periodo de retención de instantáneas
- Periodo de instantáneas
- Periodo de mantenimiento
- Tema para la notificación de SNS

6. Elija Guardar cambios.

También puede ir a la página de detalles del clúster y hacer clic en modificar para realizar modificaciones en el clúster. Si desea modificar secciones específicas del clúster, puede ir a la pestaña correspondiente de la página de detalles del clúster y hacer clic en Modificar.

Uso de la AWS CLI

Puede modificar un clúster existente con la operación AWS CLI de `update-cluster`. Para modificar un valor de configuración de un clúster, especifique el ID del clúster, el parámetro que desea cambiar y el nuevo valor del parámetro. El siguiente ejemplo cambia el periodo de mantenimiento de un clúster denominado `my-cluster` y aplica el cambio inmediatamente.

Para Linux, macOS o Unix:

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --preferred-maintenance-window sun:23:00-mon:02:00
```

Para Windows:

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
    --preferred-maintenance-window sun:23:00-mon:02:00
```

Para obtener más información, consulte update-cluster en la Referencia de comandos de la AWS CLI.

Uso de la API de MemoryDB

Puede modificar un clúster existente con la operación de la API de MemoryDB UpdateCluster. Para modificar un valor de configuración de un clúster, especifique el ID del clúster, el parámetro que desea cambiar y el nuevo valor del parámetro. El siguiente ejemplo cambia el periodo de mantenimiento de un clúster denominado `my-cluster` y aplica el cambio inmediatamente.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=UpdateCluster
    &ClusterName=my-cluster
    &PreferredMaintenanceWindow=sun:23:00-mon:02:00
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210801T220302Z
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210802T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Agregar/eliminar nodos de un clúster

Puede agregar o eliminar nodos de un clúster mediante la AWS Management Console, la AWS CLI o la API de MemoryDB.

Uso de la AWS Management Console

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. En la lista de clústeres, elija el nombre del clúster del que desea agregar o eliminar un nodo.
3. En la pestaña Particiones y nodos, seleccione Agregar o eliminar nodos
4. En Nuevo número de nodos, introduzca el número de nodos que desea.
5. Seleccione Confirmar.

> **Important**
>
> Si establece el número de nodos en 1, dejará de estar habilitado para Multi-AZ. También puede optar por activar la conmutación por error automática.

Uso de la AWS CLI

1. Especifique los nombres de los nodos que desea eliminar. Para obtener más información, consulte Visualización de los detalles de un clúster.
2. Utilice la operación `update-cluster` de la CLI con una lista de los nodos que desea quitar, como en el siguiente ejemplo.

   Para quitar nodos de un clúster a través de la interfaz de línea de comandos, utilice el comando `update-cluster` con los siguientes parámetros:

   - `--cluster-name` El ID del clúster del que desea quitar nodos.
   - `--replica-configuration`: permite establecer el número de réplicas:
     - `ReplicaCount`: defina esta propiedad para especificar el número de nodos de réplica que desea.
   - `--region` especifica la región de AWS del clúster del que desea quitar nodos.

Para Linux, macOS o Unix:

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --replica-configuration \
        ReplicaCount=1 \
    --region us-east-1
```

Para Windows:

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
    --replica-configuration ^
        ReplicaCount=1 ^
    --region us-east-1
```

Para obtener más información, consulte los temas de la AWS CLI `update-cluster`.

Uso de la API de MemoryDB

Para eliminar nodos utilizando la API de MemoryDB, llame a la operación de la API `UpdateCluster` con el ID de clúster y una lista de los nodos que desea eliminar, tal y como se muestra a continuación:

- `ClusterName` El ID del clúster del que desea quitar nodos.
- `ReplicaConfiguration`: permite establecer el número de réplicas:
  - `ReplicaCount`: defina esta propiedad para especificar el número de nodos de réplica que desea.
- `Region` especifica la región de AWS del clúster del que desea quitar un nodo.

Para obtener más información, consulte UpdateCluster.

Acceso al clúster

Las instancias de MemoryDB para Redis se han diseñado para obtener acceso a ellos a través de una instancia de Amazon EC2.

Puede acceder al nodo de MemoryDB desde una instancia de Amazon EC2 en la misma Amazon VPC. O, utilizando la conexión de emparejamiento de VPC, puede obtener acceso al nodo de MemoryDB desde una Amazon EC2 en otra Amazon VPC.

Temas

Conceder acceso a su clúster

Puede conectarse al clúster de MemoryDB solo desde una instancia de Amazon EC2 que se ejecuta en la misma Amazon VPC. En este caso, necesitará conceder acceso de red al clúster.

Para conceder acceso de red desde un grupo de seguridad de Amazon VPC a un clúster

1. Inicie sesión en la AWS Management Console y abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. En el panel de navegación de la izquierda, debajo de Network & Security, elija Security Groups.
3. En la lista de grupos de seguridad, elija el grupo de seguridad para su Amazon VPC. A menos que haya creado un grupo de seguridad para que lo utilice MemoryDB, este grupo de seguridad se denominará default.
4. Elija la pestaña Inbound y haga lo siguiente:
    a. Elija Editar.
    b. Seleccione Agregar regla.
    c. En la columna Type, elija Custom TCP rule.
    d. En el cuadro Port range, escriba el número de puerto para su nodo de clúster. Este número debe ser el mismo que especificó cuando lanzó el clúster. El puerto predeterminado para Redis es **6379**.

e. En el cuadro Origen, elija Cualquier lugar que tenga el rango de puertos (0.0.0.0/0) para que cualquier instancia de Amazon EC2 que lance en su Amazon VPC pueda conectarse a sus nodos de MemoryDB.

> **Important**
>
> Al abrir el clúster de MemoryDB a 0.0.0.0/0 no se expone el clúster a Internet, ya que no tiene ninguna dirección IP pública y, por lo tanto, no se puede acceder a este desde fuera de la VPC. Sin embargo, el grupo de seguridad predeterminado se puede aplicar a otras instancias de Amazon EC2 en la cuenta del cliente y dichas instancias pueden tener una dirección IP pública. Si se está ejecutando algo en el puerto predeterminado, ese servicio podría exponerse de forma involuntaria. Por lo tanto, recomendamos crear un grupo de seguridad de VPC para que lo utilice exclusivamente MemoryDB. Para obtener más información, consulte Grupos de seguridad personalizados.

f. Seleccione Save.

Al lanzar una instancia de Amazon EC2 en su Amazon VPC, esa instancia podrá conectarse a su clúster de MemoryDB.

Acceso a los recursos de MemoryDB desde fuera de AWS

MemoryDB es un servicio diseñado para que se utilice internamente en su VPC. Se desaconseja el acceso externo debido a la latencia del tráfico de Internet y a los riesgos de seguridad. Sin embargo, si se requiere acceso externo a MemoryDB para fines de desarrollo o pruebas, puede obtenerse a través de una VPN.

Mediante la AWS Client VPN, puede permitir el acceso externo a los nodos de MemoryDB con los siguientes beneficios:

- Acceso restringido a usuarios aprobados o claves de autenticación
- Tráfico cifrado entre Client VPN y el punto de conexión de VPN de AWS
- Acceso limitado a subredes o nodos específicos
- Revocación sencilla del acceso de los usuarios o claves de autenticación
- Conexiones de auditoría

Los siguientes procedimientos muestran cómo:

Temas

- Crear una entidad de certificación
- Configuración de componentes de AWS Client VPN
- Configurar el cliente de VPN

Crear una entidad de certificación

Es posible crear una entidad de certificación (CA) utilizando diferentes técnicas o herramientas. Recomendamos la utilidad easy-rsa, proporcionada por el proyecto OpenVPN . Independientemente de la opción que elija, asegúrese de mantener protegidas las claves. El siguiente procedimiento descarga los scripts de easy-rsa, y crea la entidad de certificación y las claves para autenticar el primer cliente de VPN:

- Para crear los certificados iniciales, abra un terminal y proceda del modo siguiente:
  - `git clone` https://github.com/OpenVPN/easy-rsa
  - `cd easy-rsa`
  - `./easyrsa3/easyrsa init-pki`

- `./easyrsa3/easyrsa build-ca nopass`
- `./easyrsa3/easyrsa build-server-full server nopass`
- `./easyrsa3/easyrsa build-client-full client1.domain.tld nopass`

Se creará un subdirectorio pki que contiene los certificados bajo easy-rsa.

- Envíe el certificado del servidor a AWS Certificate manager (ACM):
  - En la consola de ACM, seleccione Certificate Manager.
  - Seleccione Importar certificado.
  - Especifique el certificado de clave pública disponible en el archivo `easy-rsa/pki/issued/server.crt` en el campo Cuerpo del certificado .
  - Pegue la clave privada disponible en `easy-rsa/pki/private/server.key` en el campo Clave privada del certificado. Asegúrese de seleccionar todas las líneas entre `BEGIN AND END PRIVATE KEY` (incluidas las líneas `END` y `BEGIN`).
  - Pegue la clave pública de CA disponible en el archivo `easy-rsa/pki/ca.crt` en el campo Cadena de certificados.
  - Seleccione Revisar e importar.
  - Seleccione Importar.

  Para enviar los certificados del servidor a ACM mediante la AWS CLI, ejecute el siguiente comando: `aws acm import-certificate --certificate fileb://easy-rsa/pki/issued/server.crt --private-key file://easy-rsa/pki/private/server.key --certificate-chain file://easy-rsa/pki/ca.crt --region` *`region`*.

  Anote el ARN del certificado para usarlo en el futuro.

Configuración de componentes de AWS Client VPN

Mediante la consola de AWS

En la consola de AWS, seleccione Servicios y, a continuación, VPC.

En Red virtual privada, seleccione Puntos de conexión de VPN de cliente y proceda del modo siguiente:

Configuración de componentes de AWS Client VPN

- Seleccione Crear punto de conexión de VPN de cliente.

- Especifique las opciones siguientes:
  - CIDR de IPv4 de cliente: utilice una red privada con una máscara de red de al menos el intervalo /22. Asegúrese de que la subred seleccionada no entra en conflicto con las direcciones de las redes de la VPC. Ejemplo 10.0.0.0/22.
  - En ARN del certificado del servidor, seleccione el ARN del certificado importado previamente.
  - Seleccione Utilizar la autenticación mutua.
  - En ARN del certificado de cliente, seleccione el ARN del certificado importado previamente.
  - Seleccione Crear punto de conexión de VPN de cliente.

Uso de AWS CLI

Ejecute el siguiente comando:

```
aws ec2 create-client-vpn-endpoint --client-cidr-block
"10.0.0.0/22" --server-certificate-arn arn:aws:acm:us-
east-1:012345678912:certificate/0123abcd-ab12-01a0-123a-123456abcdef --
authentication-options Type=certificate-
authentication,,MutualAuthentication={ClientRootCertificateChainArn=arn:aws:acm:
east-1:012345678912:certificate/123abcd-ab12-01a0-123a-123456abcdef} --
connection-log-options Enabled=false
```

Ejemplo de resultados:

```
"ClientVpnEndpointId": "cvpn-endpoint-0123456789abcdefg",
"Status": { "Code": "pending-associate" }, "DnsName": "cvpn-
endpoint-0123456789abcdefg.prod.clientvpn.us-east-1.amazonaws.com" }
```

Asociar las redes de destino al punto de conexión de VPN

- Seleccione el nuevo punto de conexión de VPN y, a continuación, seleccione la pestaña Asociaciones .
- Seleccione Asociar y especifique las siguientes opciones.
  - VPC: seleccione la VPC del clúster de MemoryDB.
  - Seleccione una de las redes del clúster de MemoryDB. En caso de duda, revise las redes en Grupos de subredes en el panel de MemoryDB.
  - Seleccione Asociar. Si es necesario, repita los pasos para las redes restantes.

Uso de AWS CLI

Ejecute el siguiente comando:

```
aws ec2 associate-client-vpn-target-network --client-vpn-endpoint-id cvpn-
endpoint-0123456789abcdefg --subnet-id subnet-0123456789abdcdef
```

Ejemplo de resultados:

```
"Status": { "Code": "associating" }, "AssociationId": "cvpn-
assoc-0123456789abdcdef" }
```

Revisar el grupo de seguridad de VPN

El punto de conexión de VPN adoptará automáticamente el grupo de seguridad predeterminado de la VPC. Compruebe las reglas de entrada y salida y confirme si el grupo de seguridad permite el tráfico desde la red VPN (definida en la opción Punto de conexión de VPN) a las redes de MemoryDB en los puertos de servicio (de forma predeterminada, 6379 para Redis).

Si necesita cambiar el grupo de seguridad asignado al punto de conexión de VPN, proceda de la siguiente manera:

- Seleccione el grupo de seguridad actual
- Seleccione Aplicar grupo de seguridad.
- Seleccione el nuevo grupo de seguridad.

Uso de AWS CLI

Ejecute el siguiente comando:

```
aws ec2 apply-security-groups-to-client-vpn-target-network --
client-vpn-endpoint-id cvpn-endpoint-0123456789abcdefga  --vpc-id
vpc-0123456789abdcdef --security-group-ids sg-0123456789abdcdef
```

Ejemplo de resultados:

```
"SecurityGroupIds": [ "sg-0123456789abdcdef" ] }
```

> **Note**
>
> El grupo de seguridad de MemoryDB también necesita permitir el tráfico procedente de los clientes de VPN. Las direcciones de los clientes se enmascararán con la dirección del punto

de conexión de VPN, de acuerdo con la red de la VPC. Por lo tanto, tenga en cuenta la red de la VPC (no la red de los clientes de VPN) cuando cree la regla de entrada en el grupo de seguridad de MemoryDB.

Autorizar el acceso de VPN a las redes de destino

En la pestaña Autorización seleccione Autorizar entrada y especifique lo siguiente:

- Red de destino para habilitar el acceso: utilice 0.0.0.0/0 para permitir el acceso a cualquier red (incluido Internet) o restrinja las redes o hosts de MemoryDB.
- En Conceder acceso a:, seleccione Permitir el acceso a todos los usuarios.
- Seleccione Añadir reglas de autorización.

Uso de AWS CLI

Ejecute el siguiente comando:

```
aws ec2 authorize-client-vpn-ingress --client-vpn-endpoint-id cvpn-
endpoint-0123456789abcdefg --target-network-cidr 0.0.0.0/0 --authorize-all-
groups
```

Ejemplo de resultados:

```
{ "Status": { "Code": "authorizing" } }
```

Permitir el acceso a Internet desde los clientes de VPN

Si necesita navegar por Internet a través de la VPN, debe crear una ruta adicional. Seleccione la pestaña Route Table (Tabla de ruteo) y, a continuación, seleccione Create Route (Crear ruta):

- Destino de la ruta: 0.0.0.0/0
- Target VPC Subnet ID (ID de subred de la VPC de destino): seleccione una de las subredes asociadas con acceso a Internet.
- Seleccione Create Route (Crear ruta).

Uso de AWS CLI

Ejecute el siguiente comando:

```
aws ec2 create-client-vpn-route --client-vpn-endpoint-id cvpn-
endpoint-0123456789abcdefg --destination-cidr-block 0.0.0.0/0 --target-vpc-
subnet-id subnet-0123456789abdcdef
```

Ejemplo de resultados:

```
{ "Status": { "Code": "creating" } }
```

Configurar el cliente de VPN

En el panel de AWS Client VPN, seleccione el punto de conexión de VPN creado recientemente y seleccione Download Client Configuration (Descargar configuración del cliente). Copie el archivo de configuración y los archivos `easy-rsa/pki/issued/client1.domain.tld.crt` y `easy-rsa/pki/private/client1.domain.tld.key`. Edite el archivo de configuración y cambie o agregue los siguientes parámetros:

- cert: agregue una nueva línea con el parámetro cert apuntando al archivo `client1.domain.tld.crt`. Utilice la ruta completa al archivo. Ejemplo: `cert /home/user/.cert/client1.domain.tld.crt`
- cert: key: agregue una nueva línea con el parámetro key apuntando al archivo `client1.domain.tld.key`. Utilice la ruta completa al archivo. Ejemplo: `key /home/user/.cert/client1.domain.tld.key`

Establezca la conexión de VPN con el comando: `sudo openvpn --config downloaded-client-config.ovpn`

Revocar el acceso

Si necesita invalidar el acceso de una clave de cliente concreta, la clave debe revocarse en la CA. A continuación, envíe la lista de revocación a AWS Client VPN.

Revocar la clave con easy-rsa:

- `cd easy-rsa`
- `./easyrsa3/easyrsa revoke client1.domain.tld`
- Especifique "yes" (sí) para continuar o escriba cualquier otra entrada para cancelar.

  ```
  Continue with revocation: `yes` ... * `./easyrsa3/easyrsa gen-crl
  ```
- Se ha creado una CRL actualizada. Archivo CRL: `/home/user/easy-rsa/pki/crl.pem`

Importar la lista de revocación a AWS Client VPN:

- En la AWS Management Console, seleccione Services (Servicios) y, a continuación, VPC.
- Seleccione Puntos de conexión de VPN de cliente.
- Seleccione el punto de conexión de Client VPN y, a continuación, seleccione Actions (Acciones) -> Import Client Certificate CRL (Importar CRL de certificado de cliente).
- Pegue el contenido del archivo `crl.pem`.

Uso de AWS CLI

Ejecute el siguiente comando:

```
aws ec2 import-client-vpn-client-certificate-revocation-list --certificate-revocation-list file://./easy-rsa/pki/crl.pem --client-vpn-endpoint-id cvpn-endpoint-0123456789abcdefg
```

Ejemplo de resultados:

```
Example output: { "Return": true }
```

Búsqueda de puntos de conexión

Su aplicación se conecta a su clúster mediante el punto de conexión. Un punto de conexión es una dirección única del clúster. Utilice el punto de conexión del clúster para todas las operaciones.

Las secciones siguientes le guiarán en el proceso de detección de los puntos de conexión que necesita.

Búsqueda del punto de conexión para un clúster de MemoryDB (AWS Management Console)

Para buscar el punto de conexión de un clúster de MemoryDB

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación, elija Clusters (clústeres).

   Aparecerá la pantalla de clústeres con una lista de clústeres. Haga clic en el clúster al que desea conectarse.
3. Para buscar el punto de conexión del clúster, elija el nombre del clúster (no el botón de opción).
4. El punto de conexión del clúster se muestra debajo de Detalles del clúster. Para copiarlo, elija el ícono copiar a la izquierda del punto de conexión.

Búsqueda del punto de conexión para un clúster de MemoryDB (CLI de AWS)

Puede usar el comando `describe-clusters` para detectar el punto de conexión de un clúster. El comando devuelve el punto de conexión del clúster.

La siguiente operación recupera el punto de conexión del clúster, que en este ejemplo se representa como un *`ejemplo`* para el clúster `mycluster`.

Devuelve la siguiente respuesta JSON:

```
aws memorydb describe-clusters \
  --cluster-name mycluster
```

Para Windows:

```
aws memorydb describe-clusters ^
   --cluster-name mycluster
```

```
{
    "Clusters": [
        {
            "Name": "my-cluster",
```

```
            "Status": "available",
            "NumberOfShards": 1,
            "ClusterEndpoint": {
                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                "Port": 6379
            },
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.4",
            "ParameterGroupName": "default.memorydb-redis6",
            "ParameterGroupStatus": "in-sync",
            "SubnetGroupName": "my-sg",
            "TLSEnabled": true,
            "ARN": "arn:aws:memorydb:us-east-1:zzzexamplearn:cluster/my-cluster",
            "SnapshotRetentionLimit": 0,
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "SnapshotWindow": "04:30-05:30",
            "ACLName": "my-acl",
            "AutoMinorVersionUpgrade": true
        }
    ]
}
```

Para obtener más información, consulte describe-clusters.

Búsqueda del punto de conexión para un clúster de MemoryDB (API de MemoryDB)

Puede usar la API de MemoryDB para Redis para detectar el punto de conexión de un clúster.

Búsqueda del punto de conexión para un clúster de MemoryDB (API de MemoryDB)

Puede usar la API de MemoryDB para detectar el punto de conexión de un clúster con la acción `DescribeClusters`. La acción devuelve el punto de conexión del clúster.

La siguiente operación recupera el punto de conexión del clúster `mycluster`.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DescribeClusters
    &ClusterName=mycluster
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210802T192317Z
    &Version=2021-01-01
    &X-Amz-Credential=<credential>
```

Para obtener más información, consulte DescribeClusters.

Trabajar con particiones

Una partición es una colección de uno a seis nodos. Puede crear un clúster con un mayor número de particiones y un menor número de réplicas con un total de hasta 500 nodos por clúster. Esta configuración de clúster puede variar desde 500 particiones y 0 réplicas hasta 100 particiones y 4 réplicas, que es el número máximo de réplicas permitido. Los datos del clúster están particionados en las distintas particiones del clúster. Si hay más de un nodo en una partición, esta implementa la reproducción con un nodo, siendo el nodo principal de lectura/escritura y los demás, nodos de réplica de solo lectura.

Cuando crea un clúster de MemoryDB usando AWS Management Console, debe especificar el número de particiones del clúster y el número de nodos de las particiones. Para obtener más información, consulte Creación de un clúster de MemoryDB.

Los nodos de las particiones tienen las mismas especificaciones de memoria, almacenamiento y computación. La API de MemoryDB le permite controlar los atributos de todo el clúster, como el número de nodos, la configuración de seguridad y los periodos de mantenimiento del sistema.

Para obtener más información, consulte Cambios en las particiones y reequilibrado de particiones sin conexión para MemoryDB y Cambios en las particiones y reequilibrado de particiones en línea para MemoryDB.

Búsqueda del nombre de una partición

Puede encontrar el nombre de una partición utilizando la AWS Management Console, la AWS CLI o la API de MemoryDB..

Utilización de la AWS Management Console

El siguiente procedimiento utiliza la AWS Management Console para buscar los nombres de las particiones de un clúster de MemoryDB.

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación izquierdo, elija Clústeres.
3. Elija el clúster en Nombre cuyos nombres de particiones desee buscar.
4. En la pestaña Particiones y nodos, consulte la lista de particiones en Nombre. También puede ampliar cada uno de ellos para ver los detalles de sus nodos.

Utilización de la AWS CLI

Para encontrar los nombres de las particiones (particiones) para los clústeres de MemoryDB, utilice la operación de la AWS CLI `describe-clusters` con el siguiente parámetro opcional.

- **`--cluster-name`**: un parámetro opcional que, cuando se utiliza, limita los resultados a los detalles del clúster especificado. Si se omite este parámetro, se devuelven los detalles de hasta 100 clústeres.
- **`--show-shard-details`**: devuelve los detalles de las particiones, incluidos sus nombres.

Este comando devuelve los detalles de `my-cluster`.

Para Linux, macOS o Unix:

```
aws memorydb describe-clusters \
    --cluster-name my-cluster
```

```
    --show-shard-details
```

Para Windows:

```
aws memorydb describe-clusters ^
    --cluster-name my-cluster
    --show-shard-details
```

Devuelve la siguiente respuesta JSON:

Se agregan saltos de línea para facilitar la lectura.

```
{
    "Clusters": [
        {
            "Name": "my-cluster",
            "Status": "available",
            "NumberOfShards": 1,
            "Shards": [
                {
                    "Name": "0001",
                    "Status": "available",
                    "Slots": "0-16383",
                    "Nodes": [
                        {
                            "Name": "my-cluster-0001-001",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        },
                        {
                            "Name": "my-cluster-0001-002",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1b",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
```

```
                                "Address": "clustercfg.my-cluster.xxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        }
                    ],
                    "NumberOfNodes": 2
                }
            ],
            "ClusterEndpoint": {
                "Address": "clustercfg.my-cluster.xxxxx.memorydb.us-
east-1.amazonaws.com",
                "Port": 6379
            },
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "ParameterGroupName": "default.memorydb-redis6",
            "ParameterGroupStatus": "in-sync",
            "SubnetGroupName": "my-sg",
            "TLSEnabled": true,
            "ARN": "arn:aws:memorydb:us-east-1:xxxxxexamplearn:cluster/my-cluster",
            "SnapshotRetentionLimit": 0,
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "SnapshotWindow": "04:30-05:30",
            "ACLName": "my-acl",
            "DataTiering": "false",
            "AutoMinorVersionUpgrade": true
        }
    ]
}
```

Uso de la API de MemoryDB

Para encontrar los identificadores de particiones para los clústeres de MemoryDB, utilice la operación de API `DescribeClusters` con el siguiente parámetro opcional.

- **`ClusterName`**: un parámetro opcional que, cuando se utiliza, limita los resultados a los detalles del clúster especificado. Si se omite este parámetro, se devuelven los detalles de hasta 100 clústeres.
- **`ShowShardDetails`**: devuelve los detalles de las particiones, incluidos sus nombres.

Example

Este comando devuelve los detalles de `my-cluster`.

Para Linux, macOS o Unix:

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DescribeClusters
   &ClusterName=sample-cluster
   &ShowShardDetails=true
   &Version=2021-01-01
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20210802T192317Z
   &X-Amz-Credential=<credential>
```

Administrar la implementación de MemoryDB

En esta sección, encontrará información acerca de cómo administrar los diferentes componentes de la implementación de MemoryDB.

Temas

Versiones del motor de Redis

En esta sección, se detallan las versiones compatibles del motor de Redis.

Temas

MemoryDB para Redis versión 7.1 (mejorada)

La versión 7.1 de MemoryDB para Redis añade compatibilidad con las funciones de búsqueda vectorial en una versión preliminar para determinadas regiones, así como correcciones de errores críticos y mejoras de rendimiento.

- Función de búsqueda vectorial: la búsqueda vectorial se puede utilizar con la funcionalidad existente de MemoryDB. Las aplicaciones que no utilizan la búsqueda vectorial no se verán afectadas por su presencia. La vista previa de la búsqueda vectorial está disponible en MemoryDB para Redis a partir de la versión 7.1 en las siguientes regiones: EE.UU. Este (Norte de Virginia y Ohio), EE.UU. Oeste (Oregón), UE (Irlanda) y Asia Pacífico (Tokio). Consulte la documentación aquí para saber cómo activar la vista previa de la búsqueda vectorial y las funciones relacionadas.

> **Note**
>
> La versión 7.1 de MemoryDB para Redis es compatible con OSS Redis v7.0. Para obtener más información sobre la versión 7.0 de Redis, consulte las notas de la versión 7.0 de Redis en Redis on. GitHub

MemoryDB para Redis versión 7.0 (mejorada)

MemoryDB para Redis 7.0 agrega una serie de mejoras y compatibilidad con nuevas funciones:

- Funciones de Redis: MemoryDB para Redis 7 agrega compatibilidad para funciones de Redis y proporciona una experiencia administrada que permite a los desarrolladores ejecutar scripts de LUA con la lógica de la aplicación almacenada en el clúster de MemoryDB, sin necesidad de que los clientes vuelvan a enviar los scripts al servidor con cada conexión.
- Mejoras de ACL: MemoryDB para Redis 7 agrega compatibilidad para la próxima versión de las listas de control de acceso (ACL) de Redis. Con MemoryDB para Redis 7, los clientes ahora pueden especificar varios conjuntos de permisos en claves o espacios de claves específicos en Redis.
- Publicación/envío particionado: MemoryDB para Redis 7 agrega compatibilidad para ejecutar la funcionalidad de publicación/envío de Redis de forma particionada cuando se ejecuta MemoryDB en modo de clúster habilitado (CME). Las capacidades de publicación/envío de Redis permiten a los editores emitir mensajes a cualquier número de suscriptores de un canal. Con Amazon MemoryDB para Redis 7, los canales se enlazan a una partición del clúster de MemoryDB, lo que elimina la necesidad de propagar la información del canal entre las particiones. Esto se traduce en una escalabilidad mejorada.
- Multiplexación de E/S mejorada: la versión 7 de MemoryDB para Redis incluye una multiplexación de E/S mejorada, que ofrece un mayor rendimiento y una menor latencia para cargas de trabajo de alto rendimiento que tienen muchas conexiones de cliente simultáneas a un clúster de MemoryDB.

Por ejemplo, al utilizar un clúster de nodos r6g.4xlarge y ejecutar 5200 clientes simultáneos, puede lograr un aumento de hasta un 46 % en el rendimiento (operaciones de lectura y escritura por segundo) y una disminución de la latencia de P99 de hasta un 21 %, en comparación con la versión 6 de MemoryDB para Redis.

Para obtener más información sobre la versión 7.0 de Redis, consulte las notas de la versión 7.0 de Redis en Redis on. GitHub

MemoryDB para Redis versión 6.2 (mejorada)

MemoryDB presenta la próxima versión del motor de Redis, que incluye soporte de actualización automática de versiones Autenticación de usuarios con listas de control de acceso (ACL), almacenamiento en caché del lado del cliente y mejoras operativas significativas.

La versión 6.2.6 del motor Redis también incluye la compatibilidad con el formato nativo de notación de JavaScript objetos (JSON), una forma sencilla y sin esquemas de codificar conjuntos de datos complejos dentro de los clústeres de Redis. Con la compatibilidad con JSON, puede aprovechar el rendimiento y las API de Redis para las aplicaciones que funcionan a través de JSON. Para obtener más información, consulte Introducción a JSON. También se incluye una métrica relacionada con JSON `JsonBasedCmds` que se incorpora para monitorear el uso de este tipo de datos. CloudWatch Para obtener más información, consulte Métricas de MemoryDB.

Con Redis 6, MemoryDB ofrecerá una sola versión para cada versión secundaria de Redis OSS, en lugar de ofrecer varias versiones de parche. Se ha diseñado para minimizar la confusión y la ambigüedad al tener que elegir entre varias versiones secundarias. MemoryDB también administrará automáticamente la versión secundaria y la versión del parche de los clústeres en ejecución, lo que garantiza un mejor rendimiento y mayor seguridad. Esto se gestionará a través de canales de notificación estándar a los clientes mediante una campaña de actualización de servicio. Para obtener más información, consulte Actualizaciones de los servicios de MemoryDB para Redis.

Si no especifica la versión del motor durante la creación, MemoryDB seleccionará automáticamente la versión de Redis que prefiera. Por otro lado, si especifica la versión del motor mediante el uso de `6.2`, MemoryDB invocará automáticamente la versión de parche preferida de Redis 6.2 que se encuentre disponible.

Por ejemplo, al crear un clúster, establece la propiedad del parámetro `--engine-version` en `6.2`. El clúster se lanzará con la versión de parche preferida actual disponible en el momento de creación. Cualquier solicitud con un valor de la versión de motor completa se rechazará, se lanzará una excepción y el proceso fallará.

Al llamar a la API `DescribeEngineVersions`, el valor de parámetro `EngineVersion` se establecerá en 6.2 y la versión real del motor se devolverá en el campo `EnginePatchVersion`.

Para obtener más información sobre la versión 6.2 de Redis, consulte las notas de la versión 6.2 de Redis en Redis on. GitHub

Actualización de las versiones del motor

De forma predeterminada, MemoryDB administra automáticamente la versión de parche de los clústeres en ejecución mediante actualizaciones de servicio. También puede inhabilitar la actualización automática de la versión secundaria si establece la propiedad `AutoMinorVersionUpgrade` de sus clústeres en false. Sin embargo, no puede excluirse de la actualización de la versión de parches automáticos.

Puede controlar si se actualiza el software compatible con los protocolos de su clúster a nuevas versiones compatibles con MemoryDB, así como el momento en que se realizan dichas actualizaciones, antes de que comience la actualización automática. Este nivel de control permite mantener la compatibilidad con versiones concretas, probar nuevas versiones con la aplicación antes de implementarlas en producción y realizar actualizaciones de versiones en los horarios y los plazos que más le convengan.

Puede iniciar las actualizaciones de las versiones del motor en su clúster de las siguientes maneras:

- Actualizándolo y especificando una nueva versión del motor. Para obtener más información, consulte Modificación de un clúster de MemoryDB.
- Aplicando la actualización del servicio a la versión de motor correspondiente. Para obtener más información, consulte Actualizaciones de los servicios de MemoryDB para Redis.

Tenga en cuenta lo siguiente:

- Puede actualizar a una versión de motor más reciente, pero no puede volver a una versión de motor más antigua. Si desea usar una versión de motor más antigua, deberá eliminar el clúster existente y crearlo de nuevo con la versión del motor más antigua.
- Recomendamos actualizar periódicamente a la última versión principal, ya que la mayoría de las mejoras importantes no se transfieren a versiones anteriores. A medida que MemoryDB amplía la disponibilidad a una nueva AWS región, MemoryDB es compatible con las dos `MAJOR.MINOR` versiones más recientes de la nueva región en ese momento. Por ejemplo, si se lanza una nueva AWS región y las versiones más recientes de `MAJOR.MINOR` MemoryDB para Redis son 7.0 y

6.2, MemoryDB for Redis admitirá las versiones 7.0 y 6.2 en la nueva región. AWS A medida que se publiquen nuevas versiones `MAJOR.MINOR` de MemoryDB para Redis, MemoryDB continuará añadiendo soporte para las nuevas versiones de MemoryDB for Redis. Para obtener más información sobre cómo elegir Regions para MemoryDB, consulte Regiones y puntos de conexión admitidos.

- La administración de la versión del motor está diseñada para que pueda tener el mayor control posible sobre cómo se produce la aplicación de parches. Sin embargo, MemoryDB se reserva el derecho de aplicar en su nombre el parche a su clúster en el caso improbable de una vulnerabilidad de seguridad crítica en el sistema o el software.
- MemoryDB ofrecerá una sola versión para cada versión secundaria de Redis OSS, en lugar de ofrecer varias versiones de parche. Se ha diseñado para minimizar la confusión y la ambigüedad al tener que elegir entre varias versiones. MemoryDB también administrará automáticamente la versión secundaria y la versión del parche de los clústeres en ejecución, lo que garantiza un mejor rendimiento y mayor seguridad. Esto se gestionará a través de canales de notificación estándar a los clientes mediante una campaña de actualización de servicio. Para obtener más información, consulte Actualizaciones de los servicios de MemoryDB para Redis.
- Puede actualizar la versión del clúster con un tiempo de inactividad mínimo. El clúster está disponible para operaciones de lectura durante toda la actualización y para operaciones de escritura durante la mayoría del proceso, excepto durante la operación de conmutación por error, que dura unos segundos.
- Se recomienda que actualice el motor durante los periodos de poco tráfico entrante.

  Los clústeres con varias particiones se procesan y se aplican parches de la siguiente manera:

  - Solo se realiza una operación de actualización en una partición a la vez.
  - En cada partición, todas las réplicas se procesan antes que el principal. Si hay menos réplicas en una partición, el principal de esa partición podrá procesarse antes que las réplicas de otras particiones terminen de procesarse.
  - En todas las particiones, los nodos principales se procesan en series. Solo se actualiza un nodo principal a la vez.

Temas

- Cómo actualizar las versiones del motor
- Resolución de actualizaciones del motor de Redis bloqueadas

Cómo actualizar las versiones del motor

Para iniciar las actualizaciones de versión de su clúster, debe modificarlo mediante la consola de MemoryDB, la API de MemoryDB o la API de MemoryDB y AWS CLI especificando una versión de motor más reciente. Para obtener más información, consulte los siguientes temas.

- Uso de la AWS Management Console
- Uso de la AWS CLI
- Uso de la API de MemoryDB

Resolución de actualizaciones del motor de Redis bloqueadas

Tal y como se muestra en la siguiente tabla, la operación de actualización del motor de Redis se bloqueará si tiene una operación de ampliación pendiente.

Operaciones pendientes	Operaciones bloqueadas
Escalado ascendente	Actualización del motor inmediata
Actualización del motor	Ampliación inmediata
Ampliación y actualización del motor	Ampliación inmediata
	Actualización del motor inmediata

Introducción a JSON

MemoryDB admite el formato nativo de JavaScript Object Notation (JSON, Notación de objetos de JavaScript), una forma sencilla y sin esquemas de codificar conjuntos de datos complejos dentro de clústeres de Redis. Puede almacenar datos de forma nativa y acceder a ellos utilizando el formato JSON dentro de clústeres de Redis, así como actualizar los datos JSON almacenados en esos clústeres, sin necesidad de administrar un código personalizado para serializarlo y deserializarlo.

Además de usar operaciones del API de Redis para las aplicaciones que funcionan sobre JSON, ahora puede recuperar y actualizar de manera eficiente partes específicas de un documento JSON sin necesidad de manipular todo el objeto, lo que puede mejorar el rendimiento y reducir el

costo. También puede buscar en el contenido de su documento JSON mediante la consulta Estilo GoessnerJSONPath.

Después de crear un clúster con una versión de motor compatible, el tipo de datos JSON y los comandos asociados están disponibles automáticamente. Esta es una API compatible y una RDB compatible con la versión 2 del módulo RedisJSON, por lo que puede migrar fácilmente las aplicaciones Redis existentes basadas en JSON a MemoryDB. Para obtener más información acerca de los comandos de Redis compatibles de la, consulte Comandos admitidos.

La métrica relacionada con JSON `JsonBasedCmds` se incorpora en CloudWatch para monitorear el uso de este tipo de datos. Para obtener más información, consulte Métricas de MemoryDB.

> **Note**
>
> Para usar JSON, debe ejecutar la versión del motor de Redis 6.2.6 o posterior.

Temas

- Información general del tipo de datos JSON de Redis
- Comandos admitidos

Información general del tipo de datos JSON de Redis

MemoryDB admite una serie de comandos de Redis para trabajar con el tipo de datos JSON. A continuación, se presenta información general de los datos JSON y una lista detallada de los comandos de Redis compatibles.

Terminología

Plazo	Descripción
Documento JSON	hace referencia al valor de una clave JSON de Redis..
Valor JSON	hace referencia a un subconjunto de un JSON, incluida la raíz que representa a todo el documento. Un valor podría ser un contenedor o una entrada dentro de un contenedor

Plazo	Descripción
Elemento JSON	equivalente al valor JSON.

Estándares JSON admitidos

El formato JSON es compatible con el estándar de intercambio de datos JSON RFC 7159 y ECMA-404. Se admite UTF-8 Unicode en texto JSON.

Elemento raíz

El elemento raíz puede ser de cualquier tipos de datos de JSON. Tenga en cuenta que en la RFC 4627 anterior, solo se permitían objetos o matrices como valores raíz. Desde la actualización a RFC 7159, la raíz de un documento JSON puede ser de cualquier tipo de datos JSON.

Límite de tamaño del documento

Los documentos JSON se almacenan de manera interna en un formato que se optimiza para lograr un acceso y modificación rápidos. Este formato suele consumir algo más de memoria que la representación serializada equivalente del mismo documento. El consumo de memoria de un solo documento JSON está limitado a 64 MB, que es el tamaño de la estructura de datos en memoria, no la cadena JSON. La cantidad de memoria que consume un documento JSON puede examinarse mediante el uso del comando `JSON.DEBUG MEMORY`.

JSON ACL

- El tipo de datos JSON está totalmente integrado en la capacidad Lista de control de acceso (ACL) de Redis. Similar a las categorías existentes por tipo de datos (@string, @hash, etc.), se agrega una nueva categoría @json para simplificar la administración del acceso a los comandos y datos JSON. Ningún otro comando de Redis existente es miembro de la categoría @json. Todos los comandos JSON aplican cualquier restricción y permiso de espacio de teclas o comandos.
- Hay cinco categorías de ACL de Redis existentes que se actualizan para incluir los nuevos comandos JSON: @read, @write, @fast, @slow y @admin. La tabla a continuación indica la asignación de los comandos JSON a las categorías apropiadas.

ACL

Comando JSON	@read	@write	@fast	@slow	@admin
JSON.ARRAPPEND		y	y		
JSON.ARRINDEX	y		y		
JSON.ARRINSERT		y	y		
JSON.ARRLEN	y		y		
JSON.ARRPOP		y	y		
JSON.ARRTRIM		y	y		
JSON.CLEAR		y	y		
JSON.DEBUG	y			y	y
JSON.DEL		y	y		
JSON.FORGET		y	y		
JSON.GET	y		y		
JSON.MGET	y		y		
JSON.NUMINCRBY		y	y		

Comando JSON	@read	@write	@fast	@slow	@admin
JSON.NUMMULTBY		y	y		
JSON.OBJKEYS	y		y		
JSON.OBJLEN	y		y		
JSON.RESP	y		y		
JSON.SET		y		y	
JSON.STRAPPEND		y	y		
JSON.STRLEN	y		y		
JSON.STRLEN	y		y		
JSON.TOGGLE		y	y		
JSON.TYPE	y		y		
JSON.NUMINCRBY		y	y		

Límite de profundidad de anidado

Cuando un objeto o matriz JSON tiene un elemento que es otro objeto o matriz JSON, se dice que ese objeto o matriz interior se “anida” dentro del objeto o matriz exterior. El límite máximo de profundidad de anidamiento es 128. Cualquier intento de crear un documento que contenga una profundidad de anidamiento superior a 128 se rechazará con un error.

Sintaxis de comandos

La mayoría de los comandos requieren un nombre de clave de Redis como primer argumento. Algunos comandos también tienen un argumento ruta. El argumento de ruta se establece de manera predeterminada en la raíz si es opcional y no se proporciona.

Notación:

- Los argumentos obligatorios se incluyen entre corchetes angulares, ej. <clave>
- Los argumentos opcionales deben ir entre corchetes, ej. [ruta]
- Los argumentos opcionales adicionales se indican con..., por ejemplo, [json...]

Sintaxis de ruta

Redis JSON admite dos tipos de sintaxis de rutas:

- Sintaxis mejorada: sigue la sintaxis JSONPath descrita por Goessner, como se muestra en la siguiente tabla. Hemos reordenado y modificado las descripciones de la tabla para mayor claridad.
- Sintaxis restringida: tiene capacidades de consulta limitadas.

> **Note**
>
> Los resultados de algunos comandos son sensibles al tipo de sintaxis de ruta que se utiliza.

Si una ruta de consulta comienza por '$', utiliza la sintaxis mejorada. De lo contrario, se utiliza la sintaxis restringida.

Sintaxis mejorada

Símbolo o expresión	Descripción
$	el elemento raíz
. o bien []	operador secundario
..	descenso recursivo

Símbolo o expresión	Descripción
*	comodín Todos los elementos de un objeto o matriz.
[]	operador de subíndice de matriz El índice se basa en 0.
[,]	operador de unión
[start:end:step]	operador de segmento de la matriz
?()	aplica una expresión de filtro (script) a la matriz u objeto actual
()	expresión de filtro
@	se usa en expresiones de filtro que hacen referencia al nodo actual que se está procesando
==	igual a, se utiliza en las expresiones de filtro.
!=	no es igual a, se utiliza en las expresiones de filtro.
>	mayor que, se utiliza en las expresiones de filtro.
>=	mayor o igual que, se utiliza en las expresiones de filtro.
<	menor que, se utiliza en expresiones de filtro.
<=	menor o igual que, se utiliza en las expresiones de filtro.
&&	Y lógico, se utiliza para combinar varias expresiones de filtro.

Símbolo o expresión	Descripción
||	O lógico, se utiliza para combinar varias expresiones de filtro.

Ejemplos

Los siguientes ejemplos se basan en los datos XML del ejemplo de Goessner, que hemos modificado agregando matrices adicionales.

```
{ "store": {
    "book": [
      { "category": "reference",
        "author": "Nigel Rees",
        "title": "Sayings of the Century",
        "price": 8.95,
        "in-stock": true,
        "sold": true
      },
      { "category": "fiction",
        "author": "Evelyn Waugh",
        "title": "Sword of Honour",
        "price": 12.99,
        "in-stock": false,
        "sold": true
      },
      { "category": "fiction",
        "author": "Herman Melville",
        "title": "Moby Dick",
        "isbn": "0-553-21311-3",
        "price": 8.99,
        "in-stock": true,
        "sold": false
      },
      { "category": "fiction",
        "author": "J. R. R. Tolkien",
        "title": "The Lord of the Rings",
        "isbn": "0-395-19395-8",
        "price": 22.99,
        "in-stock": false,
        "sold": false
      }
```

```
    ],
    "bicycle": {
      "color": "red",
      "price": 19.95,
      "in-stock": true,
      "sold": false
    }
  }
}
```

Ruta	Descripción
$.store.book[*].author	los autores de todos los libros de la tienda
$..author	todos los autores
$.store.*	todos los miembros de la tienda
$.store.*	todos los miembros de la tienda
$.store..price	el precio de todo lo que hay en la tienda
$..*	todos los miembros recursivos de la estructura JSON
$..book[*]	todos los libros
$..book[0]	el primer libro
$..book[-1]	el último libro
$..book[0:2]	los dos primeros libros
$..book[0,1]	los dos primeros libros
$..book[0:4]	los libros del índice 0 al 3 (el índice final no está incluido)
$..book[0:4:2]	los libros en el índice 0, 2
$..book[?(@.isbn)]	todos los libros con un número de isbn

Ruta	Descripción
$..book[?(@.price<10)]	todos los libros que cuestan menos de 10 dólares
'$..book[?(@.price < 10)]'	todos los libros que cuestan menos de 10 dólares. (La ruta debe estar entre comillas si contiene espacios en blanco).
'$..book[?(@["price"] < 10)]'	todos los libros que cuestan menos de 10 dólares
'$..book[?(@.["price"] < 10)]'	todos los libros que cuestan menos de 10 dólares
$..book[?(@.price>=10&&@.price<=100)]	todos los libros en el rango de precios de 10 a 100 dólares, incluidos
'$..book[?(@.price>=10 && @.price<=100)]'	todos los libros en el rango de precios de 10 a 100 dólares, incluidos. (La ruta debe estar entre comillas si contiene espacios en blanco).
$..book[?(@.sold==true||@.in-stock==false)]	todos los libros vendidos o agotados
'$..book[?(@.sold == true || @.in-stock == false)]'	todos los libros vendidos o agotados. (La ruta debe estar entre comillas si contiene espacios en blanco).
'$.store.book[?(@.["category"] == "fiction")]'	todos los libros de la categoría Ficción
'$.store.book[?(@.["category"] != "fiction")]'	todos los libros de las categorías que no sean Ficción

Más ejemplos de expresiones de filtro:

```
127.0.0.1:6379> JSON.SET k1 . '{"books": [{"price":5,"sold":true,"in-
stock":true,"title":"foo"}, {"price":15,"sold":false,"title":"abc"}]}'
OK
127.0.0.1:6379> JSON.GET k1 $.books[?(@.price>1&&@.price<20&&@.in-stock)]
```

```
"[{\"price\":5,\"sold\":true,\"in-stock\":true,\"title\":\"foo\"}]"
127.0.0.1:6379> JSON.GET k1 '$.books[?(@.price>1 && @.price<20 && @.in-stock)]'
"[{\"price\":5,\"sold\":true,\"in-stock\":true,\"title\":\"foo\"}]"
127.0.0.1:6379> JSON.GET k1 '$.books[?((@.price>1 && @.price<20) && (@.sold==false))]'
"[{\"price\":15,\"sold\":false,\"title\":\"abc\"}]"
127.0.0.1:6379> JSON.GET k1 '$.books[?(@.title == "abc")]'
[{"price":15,"sold":false,"title":"abc"}]

127.0.0.1:6379> JSON.SET k2 . '[1,2,3,4,5]'
127.0.0.1:6379> JSON.GET k2 $.*.[?(@>2)]
"[3,4,5]"
127.0.0.1:6379> JSON.GET k2 '$.*.[?(@ > 2)]'
"[3,4,5]"

127.0.0.1:6379> JSON.SET k3 . '[true,false,true,false,null,1,2,3,4]'
OK
127.0.0.1:6379> JSON.GET k3 $.*.[?(@==true)]
"[true,true]"
127.0.0.1:6379> JSON.GET k3 '$.*.[?(@ == true)]'
"[true,true]"
127.0.0.1:6379> JSON.GET k3 $.*.[?(@>1)]
"[2,3,4]"
127.0.0.1:6379> JSON.GET k3 '$.*.[?(@ > 1)]'
"[2,3,4]"
```

Sintaxis restringida

Símbolo o expresión	Descripción
. o bien []	operador secundario
[]	operador de subíndice de matriz El índice se basa en 0.

Ejemplos

Ruta	Descripción
.store.book[0].author	el autor del primer libro

Ruta	Descripción
.store.book[-1].author	el autor del último libro
.address.city	nombre de la ciudad
["store"]["book"][0]["title"]	el título del primer libro
["store"]["book"][-1]["title"]	el título del último libro

> **Note**
>
> Todo el contenido de Goessner citado en esta documentación está sujeto a la Licencia de Creative Commons.

Prefijos comunes de errores

Cada mensaje de error tiene un prefijo. A continuación se muestra una lista de prefijos comunes de errores:

Prefix	Descripción
ERR	un error general
LIMIT	Se ha superado el error de tamaño máximo. Por ejemplo, se ha superado el límite de tamaño del documento o el límite de profundidad de anidado
INEXISTENTE	una clave o ruta no existe
FUERA DE LOS LÍMITES	un índice de matrices fuera de los límites
SYNTAXERR	error de sintaxis
WRONGTYPE	tipo de valor incorrecto

Métricas relacionadas con JSON

Se proporcionan las siguientes métricas de información JSON:

Información	Descripción
json_total_memory_bytes	memoria total asignada a objetos JSON
json_num_documents	el número total de documentos en Redis

Para consultar las métricas principales, ejecute el comando de Redis:

```
info json_core_metrics
```

Cómo interactúa MemoryDB con JSON

A continuación, se ilustra cómo interactúa MemoryDB con el tipo de datos JSON.

Jerarquía de los operadores

Al evaluar las expresiones condicionales para el filtrado, las &&s tienen prioridad y, a continuación, se evalúan las ||s, como es común en la mayoría de los idiomas. Las operaciones entre paréntesis se ejecutarán primero.

Comportamiento del límite máximo de anidación

El límite de anidación de la ruta máxima de MemoryDB es 128. Así que un valor como `$.a.b.c.d...` solo puede alcanzar 128 niveles.

Administración de valores numéricos

JSON no tiene tipos de datos separados para números enteros y de coma flotante. Todos se llaman números.

Cuando se recibe un número JSON, se almacena en uno de dos formatos. Si el número cabe en un entero con signo de 64 bits, se convierte a ese formato; de lo contrario, se almacena como una cadena. Las operaciones aritméticas en dos números JSON (por ejemplo, JSON.NUMINCRBY y JSON.NUMMULTBY) intentan conservar la mayor precisión posible. Si los dos operandos y el valor resultante caben en un entero con signo de 64 bits, se realiza la aritmética de enteros. De lo

contrario, los operandos de entrada se convierten en números de coma flotante de doble precisión según el IEEE de 64 bits, se realiza la operación aritmética y el resultado se convierte de nuevo en una cadena.

Comandos aritméticos `NUMINCRBY` y `NUMMULTBY`:

- Si ambos números son números enteros y el resultado está fuera del rango de int64, se convierte automáticamente en un número de punto flotante de doble precisión.
- Si al menos uno de los números es un número de punto flotante, el resultado es un número de punto flotante de doble precisión.
- Si el resultado supera el rango de doble, el comando devolverá un error `OVERFLOW`.

> **Note**
>
> Antes de la versión 6.2.6.R2 del motor de Redis, cuando se recibía un número JSON en la entrada, se convertía a una de las dos representaciones binarias internas: un número entero con signo de 64 bits o un número de punto flotante de doble precisión IEEE de 64 bits. No se retiene la cadena original ni nada de su formato. Por lo tanto, cuando se genera un número como parte de una respuesta JSON, se convierte de la representación binaria interna a una cadena imprimible que utiliza reglas de formato genérico. Estas reglas podrían dan como resultado que se genere una cadena diferente de la que se recibió.
>
> - Si ambos números son números enteros y el resultado está fuera del rango de `int64`, automáticamente se convierte en un número IEEE de punto flotante de doble precisión de 64 bits.
> - Si al menos uno de los números es un punto flotante, el resultado es un número IEEE de punto flotante de doble precisión de 64 bits.
> - Si el resultado supera el rango de doble IEEE de 64 bits, el comando regresa un error `OVERFLOW`.

Para obtener una lista de los comandos disponibles, consulte el Comandos admitidos.

Evaluación de sintaxis estricta

MemoryDB no permite rutas JSON con sintaxis no válida, incluso si un subconjunto de la ruta contiene una ruta válida. Esto es para mantener un comportamiento correcto para nuestros clientes.

Comandos admitidos

Se admiten los siguientes comandos JSON de Redis:

Temas

JSON.ARRAPPEND

Adjunta uno o varios valores a los valores de la matriz en la ruta.

Sintaxis

```
JSON.ARRAPPEND <key> <path> <json> [json ...]
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (obligatoria): ruta JSON
- json (obligatorio): valor JSON que se agregará a la matriz

Devolución

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros que representan la nueva longitud de la matriz en cada ruta.
- Si un valor no es una matriz, su valor devuelto correspondiente es nulo.
- Error `SYNTAXERR` si uno de los argumentos json de entrada no es una cadena JSON válida.
- Error `NONEXISTENT` si la ruta no existe.

Si la ruta es de sintaxis restringida:

- Entero, la nueva longitud de la matriz.
- Si se seleccionan varios valores de matriz, el comando devuelve la nueva longitud de la última matriz actualizada.
- Error `WRONGTYPE` si el valor de la ruta no es una matriz.
- Error `SYNTAXERR` si uno de los argumentos json de entrada no es una cadena JSON válida.
- Error `NONEXISTENT` si la ruta no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRAPPEND  k1 $[*] '"c"'
1) (integer) 1
2) (integer) 2
```

```
3) (integer) 3
127.0.0.1:6379> JSON.GET k1
"[[\"c\"],[\"a\",\"c\"],[\"a\",\"b\",\"c\"]]"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRAPPEND  k1 [-1] '"c"'
(integer) 3
127.0.0.1:6379> JSON.GET k1
"[[],[\"a\"],[\"a\",\"b\",\"c\"]]"
```

JSON.ARRINDEX

Busca la primera aparición de un valor JSON escalar en las matrices de la ruta.

- Los errores fuera de rango se tratan redondeando el índice al principio y al final de la matriz.
- Si inicio > fin, devuelve -1 (no encontrado).

Sintaxis

```
JSON.ARRINDEX <key> <path> <json-scalar> [start [end]]
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (obligatoria): ruta JSON
- json-scalar (obligatorio): valor escalar que se debe buscar; el escalar JSON hace referencia a valores que no son objetos ni matrices. Es decir, String, number, booleano y null son valores escalares.
- inicio (opcional): índice de inicio, inclusivo. Toma 0 como valor predeterminado si no se proporciona.
- final (opcional): índice de final, exclusivo. Toma 0 como valor predeterminado si no se proporciona, lo que significa que se incluye el último elemento. 0 o -1 significa que se incluye el último elemento.

Devolución

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros. Cada valor es el índice del elemento coincidente de la matriz en la ruta. El valor es -1 si no se encuentra.
- Si un valor no es una matriz, su valor devuelto correspondiente es nulo.

Si la ruta es de sintaxis restringida:

- Entero, el índice del elemento coincidente o -1 si no se encuentra.
- Error `WRONGTYPE` si el valor de la ruta no es una matriz.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"], ["a", "b", "c"]]'
OK
127.0.0.1:6379> JSON.ARRINDEX k1 $[*] '"b"'
1) (integer) -1
2) (integer) -1
3) (integer) 1
4) (integer) 1
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
127.0.0.1:6379> JSON.ARRINDEX k1 .children '"Tom"'
(integer) 2
```

JSON.ARRINSERT

Inserta uno o varios valores en los valores de la matriz en la ruta antes del índice.

Sintaxis

```
JSON.ARRINSERT <key> <path> <index> <json> [json ...]
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (obligatoria): ruta JSON
- índice (obligatorio): índice de matriz antes del cual se insertan los valores.
- json (obligatorio): valor JSON que se agregará a la matriz

Devolución

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros que representan la nueva longitud de la matriz en cada ruta.
- Si un valor es una matriz vacía, su valor devuelto correspondiente es nulo.
- Si un valor no es una matriz, su valor devuelto correspondiente es nulo.
- Error `OUTOFBOUNDARIES` si el argumento índice está fuera de los límites.

Si la ruta es de sintaxis restringida:

- Entero, la nueva longitud de la matriz.
- Error `WRONGTYPE` si el valor de la ruta no es una matriz.
- Error `OUTOFBOUNDARIES` si el argumento índice está fuera de los límites.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRINSERT k1 $[*] 0 '"c"'
1) (integer) 1
2) (integer) 2
3) (integer) 3
127.0.0.1:6379> JSON.GET k1
"[[\"c\"],[\"c\",\"a\"],[\"c\",\"a\",\"b\"]]"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
```

```
OK
127.0.0.1:6379> JSON.ARRINSERT k1 . 0 '"c"'
(integer) 4
127.0.0.1:6379> JSON.GET k1
"[\"c\",[],[\"a\"],[\"a\",\"b\"]]"
```

JSON.ARRLEN

Obtiene la longitud de los valores de la matriz en la ruta.

Sintaxis

```
JSON.ARRLEN <key> [path]
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona

Devolución

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros, que representa la longitud de la matriz en cada ruta.
- Si un valor no es una matriz, su valor devuelto correspondiente es nulo.
- Es nulo si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Matriz de cadenas a granel. Cada elemento es un nombre clave del objeto.
- Entero, longitud de matriz.
- Si hay varios objetos seleccionados, el comando devuelve la longitud de la primera matriz.
- Error `WRONGTYPE` si el valor de la ruta no es una matriz.
- Error `WRONGTYPE` si la ruta no existe.
- Es nulo si la clave del documento no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[[], [\"a\"], [\"a\", \"b\"], [\"a\", \"b\", \"c\"]]'
(error) SYNTAXERR Failed to parse JSON string due to syntax error
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"], ["a", "b", "c"]]'
OK
127.0.0.1:6379> JSON.ARRLEN k1 $[*]
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 3

127.0.0.1:6379> JSON.SET k2 . '[[], "a", ["a", "b"], ["a", "b", "c"], 4]'
OK
127.0.0.1:6379> JSON.ARRLEN k2 $[*]
1) (integer) 0
2) (nil)
3) (integer) 2
4) (integer) 3
5) (nil)
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"], ["a", "b", "c"]]'
OK
127.0.0.1:6379> JSON.ARRLEN k1 [*]
(integer) 0
127.0.0.1:6379> JSON.ARRLEN k1 $[3]
1) (integer) 3

127.0.0.1:6379> JSON.SET k2 . '[[], "a", ["a", "b"], ["a", "b", "c"], 4]'
OK
127.0.0.1:6379> JSON.ARRLEN k2 [*]
(integer) 0
127.0.0.1:6379> JSON.ARRLEN k2 $[1]
1) (nil)
127.0.0.1:6379> JSON.ARRLEN k2 $[2]
1) (integer) 2
```

JSON.ARRPOP

Elimina y devuelve el elemento en el índice de la matriz. Al emerger una matriz vacía, se devuelve nulo.

Sintaxis

```
JSON.ARRPOP <key> [path [index]]
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona
- índice (opcional): posición en la matriz desde la que empezar a salir.
  - El valor predeterminado es -1 si no se proporciona, lo que significa el último elemento.
  - Un valor negativo significa la posición desde el último elemento.
  - Los índices fuera de los límites se redondean a sus respectivos límites de matriz.

Devolución

Si la ruta es de sintaxis mejorada:

- Matriz de cadenas masivas que representan valores emergentes en cada ruta.
- Si un valor es una matriz vacía, su valor devuelto correspondiente es nulo.
- Si un valor no es una matriz, su valor devuelto correspondiente es nulo.

Si la ruta es de sintaxis restringida:

- Cadena masiva, que representa el valor JSON emergente
- Es nulo si la matriz está vacía.
- Error `WRONGTYPE` si el valor de la ruta no es una matriz.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
```

```
OK
127.0.0.1:6379> JSON.ARRPOP k1 $[*]
1) (nil)
2) "\"a\""
3) "\"b\""
127.0.0.1:6379> JSON.GET k1
"[[],[],[\"a\"]]"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRPOP k1
"[\"a\",\"b\"]"
127.0.0.1:6379> JSON.GET k1
"[[],[\"a\"]]"

127.0.0.1:6379> JSON.SET k2 . '[[], ["a"], ["a", "b"]]'
OK
127.0.0.1:6379> JSON.ARRPOP k2 . 0
"[]"
127.0.0.1:6379> JSON.GET k2
"[[\"a\"],[\"a\",\"b\"]]"
```

JSON.ARRTRIM

Recorta una matriz en la ruta para que se convierta en una submatriz [inicio, fin], ambos inclusivos.

- Si la matriz está vacía, no se hace nada y se devuelve 0.
- Si el valor inicio es <0, trátelo como 0.
- Si el tamaño del valor final es >= (tamaño de la matriz), trátelo como tamaño-1.
- Si el tamaño del valor inicio >= o inicio > final, vacíe la matriz y devuelva 0.

Sintaxis

```
JSON.ARRINSERT <key> <path> <start> <end>
```

- clave (obligatorio): clave Redis del tipo de documento JSON

- ruta (obligatoria): ruta JSON
- inicio (obligatorio): índice de inicio, inclusivo.
- final (obligatorio): índice de final, inclusivo.

Devolución

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros que representan la nueva longitud de la matriz en cada ruta.
- Si un valor es una matriz vacía, su valor devuelto correspondiente es nulo.
- Si un valor no es una matriz, su valor devuelto correspondiente es nulo.
- Error `OUTOFBOUNDARIES` si un argumento de índice está fuera de los límites.

Si la ruta es de sintaxis restringida:

- Entero, la nueva longitud de la matriz.
- Es nulo si la matriz está vacía.
- Error `WRONGTYPE` si el valor de la ruta no es una matriz.
- Error `OUTOFBOUNDARIES` si un argumento de índice está fuera de los límites.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[[], ["a"], ["a", "b"], ["a", "b", "c"]]'
OK
127.0.0.1:6379> JSON.ARRTRIM k1 $[*] 0 1
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 2
   127.0.0.1:6379> JSON.GET k1
   "[[],[\"a\"],[\"a\",\"b\"],[\"a\",\"b\"]]"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
127.0.0.1:6379> JSON.ARRTRIM k1 .children 0 1
(integer) 2
127.0.0.1:6379> JSON.GET k1 .children
"[\"John\",\"Jack\"]"
```

JSON.CLEAR

Borra las matrices o un objeto de la ruta.

Sintaxis

```
JSON.CLEAR <key> [path]
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona

Devolución

- Entero, el número de contenedores borrados.
- La eliminación de una matriz u objeto vacío representa 0 contenedores borrados.

> **Note**
>
> Antes de la versión 6.2.6.R2 de Redis, la eliminación de una matriz u objeto vacío correspondía a 1 contenedor borrado.

- Al borrar un valor no contenedor, se devuelve 0.
- Si la ruta no encuentra ningún valor de matriz u objeto, el comando devuelve 0.

Ejemplos

```
127.0.0.1:6379> JSON.SET k1 . '[[], [0], [0,1], [0,1,2], 1, true, null, "d"]'
OK
127.0.0.1:6379>  JSON.CLEAR k1  $[*]
(integer) 6
```

```
127.0.0.1:6379> JSON.CLEAR k1  $[*]
(integer) 0
127.0.0.1:6379> JSON.SET k2 . '{"children": ["John", "Jack", "Tom", "Bob", "Mike"]}'
OK
127.0.0.1:6379> JSON.CLEAR k2 .children
(integer) 1
127.0.0.1:6379> JSON.GET k2 .children
"[]"
```

JSON.DEBUG

Información del informe. Los subcomandos admitidos son:

- MEMORY <clave> [ruta]: informa el uso de memoria en bytes de un valor JSON. La ruta se establece de forma predeterminada en la raíz si no se proporciona.
- PROFUNDIDAD <key>[ruta]: informa de la profundidad de ruta máxima del documento JSON.

> **Note**
>
> Este subcomando solo está disponible con la versión del motor de Redis 6.2.6.R2 o posterior.

- FIELDS <clave> [ruta]: informa el número de campos de la ruta del documento especificada. La ruta se establece de forma predeterminada en la raíz si no se proporciona. Cada valor JSON que no es de contenedor cuenta como un campo. Los objetos y las matrices cuentan de forma recursiva un campo para cada uno de los valores JSON que contienen. Cada valor de contenedor, excepto el contenedor raíz, cuenta como un campo adicional.
- AYUDA: imprime mensajes de ayuda del comando.

Sintaxis

```
JSON.DEBUG <subcommand & arguments>
```

Depende del subcomando:

MEMORIA

- Si la ruta es de sintaxis mejorada:

  - devuelve una matriz de números enteros, que representan el tamaño de memoria (en bytes) del valor JSON en cada ruta.
  - devuelve una matriz vacía si la clave de Redis no existe.
- Si la ruta es de sintaxis restringida:
  - devuelve un número entero, tamaño de memoria y el valor JSON en bytes.
  - devuelve nulo si la clave de Redis no existe.

DEPTH

- Devuelve un entero que representa la profundidad de ruta máxima del documento JSON.
- Devuelve nulo si la clave de Redis no existe.

FIELDS

- Si la ruta es de sintaxis mejorada:
  - devuelve una matriz de números enteros, que representa el número de campos de valor JSON en cada ruta.
  - devuelve una matriz vacía si la clave de Redis no existe.
- Si la ruta es de sintaxis restringida:
  - devuelve un número entero, el número de campos del valor JSON.
  - devuelve nulo si la clave de Redis no existe.

AYUDA: devuelve una serie de mensajes de ayuda.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[1, 2.3, "foo", true, null, {}, [], {"a":1, "b":2},
 [1,2,3]]'
OK
127.0.0.1:6379> JSON.DEBUG MEMORY k1 $[*]
1) (integer) 16
2) (integer) 16
3) (integer) 19
4) (integer) 16
```

```
5) (integer) 16
6) (integer) 16
7) (integer) 16
8) (integer) 50
9) (integer) 64
127.0.0.1:6379> JSON.DEBUG FIELDS k1 $[*]
1) (integer) 1
2) (integer) 1
3) (integer) 1
4) (integer) 1
5) (integer) 1
6) (integer) 0
7) (integer) 0
8) (integer) 2
9) (integer) 3
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.DEBUG MEMORY k1
(integer) 632
127.0.0.1:6379> JSON.DEBUG MEMORY k1 .phoneNumbers
(integer) 166

127.0.0.1:6379> JSON.DEBUG FIELDS k1
(integer) 19
127.0.0.1:6379> JSON.DEBUG FIELDS k1 .address
(integer) 4

127.0.0.1:6379> JSON.DEBUG HELP
1) JSON.DEBUG MEMORY <key> [path] - report memory size (bytes) of the JSON element.
 Path defaults to root if not provided.
2) JSON.DEBUG FIELDS <key> [path] - report number of fields in the JSON element. Path
 defaults to root if not provided.
3) JSON.DEBUG HELP - print help message.
```

JSON.DEL

Borra los valores JSON de la ruta de acceso de una clave de documento. Si la ruta es la raíz, equivale a eliminar la clave de Redis.

Sintaxis

```
JSON.DEL <key> [path]
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona

Devolución

- Número de elementos eliminados.
- 0 si la clave de Redis no existe.
- 0 si la ruta JSON no es válida o no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}, "e": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.DEL k1 $.d.*
(integer) 3
127.0.0.1:6379> JSOn.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[1,2,3,4,5]}"
127.0.0.1:6379> JSON.DEL k1 $.e[*]
(integer) 5
127.0.0.1:6379> JSOn.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[]}"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}, "e": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.DEL k1 .d.*
(integer) 3
127.0.0.1:6379> JSON.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[1,2,3,4,5]}"
127.0.0.1:6379> JSON.DEL k1 .e[*]
(integer) 5
127.0.0.1:6379> JSON.GET k1
"{\"a\":{},\"b\":{\"a\":1},\"c\":{\"a\":1,\"b\":2},\"d\":{},\"e\":[]}"
```

JSON.FORGET

Un alias de JSON.DEL

JSON.GET

Devuelve el formato JSON serializado en una o varias rutas.

Sintaxis

```
JSON.GET <key>
[INDENT indentation-string]
[NEWLINE newline-string]
[SPACE space-string]
[NOESCAPE]
[path ...]
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- SANGRÍA/NUEVA LÍNEA/ESPACIO (opcional): controla el formato de la cadena JSON devuelta, es decir, “impresión bonita”. El valor predeterminado de cada una es una cadena vacía. Se puede anular en cualquier combinación. Estos se pueden especificar en cualquier orden.
- SIN ESCAPE: opcional, puede estar presente para la compatibilidad con versiones anteriores y no tiene ningún otro efecto.
- ruta (opcional): cero o más rutas JSON, el valor predeterminado es la raíz si no se proporciona ninguna. Los argumentos de la ruta deben colocarse al final.

Devolución

Sintaxis de la ruta mejorada:

Si se da una ruta:

- Devuelve una cadena serializada de una matriz de valores.
- Si no selecciona ningún valor, el comando devuelve una matriz vacía.

Si se proporcionan varias rutas:

- Devuelve un objeto JSON con cadenas, en el que cada ruta es una clave.
- Si hay una sintaxis de ruta restringida y mejorada mixta, el resultado se ajusta a la sintaxis mejorada.
- Si no existe una ruta, su valor correspondiente es una matriz vacía.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.GET k1 $.address.*
"[\"21 2nd Street\",\"New York\",\"NY\",\"10021-3100\"]"
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" $.address.*
"[\n\t\"21 2nd Street\",\n\t\"New York\",\n\t\"NY\",\n\t\"10021-3100\"\n]"
127.0.0.1:6379> JSON.GET k1 $.firstName $.lastName $.age
"{\"$.firstName\":[\"John\"],\"$.lastName\":[\"Smith\"],\"$.age\":[27]}"
127.0.0.1:6379> JSON.SET k2 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}}'
OK
127.0.0.1:6379> json.get k2 $..*
"[{},{\"a\":1},{\"a\":1,\"b\":2},1,1,2]"
```

Sintaxis de la ruta restringida:

```
 127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.GET k1 .address
"{\"street\":\"21 2nd Street\",\"city\":\"New York\",\"state\":\"NY\",\"zipcode\":
\"10021-3100\"}"
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" .address
"{\n\t\"street\": \"21 2nd Street\",\n\t\"city\": \"New York\",\n\t\"state\": \"NY\",\n
\t\"zipcode\": \"10021-3100\"\n}"
127.0.0.1:6379> JSON.GET k1 .firstName .lastName .age
"{\".firstName\":\"John\",\".lastName\":\"Smith\",\".age\":27}"
```

JSON.MGET

Obtiene los comandos JSON serializados en la ruta de varias claves de documentos. Devuelve un valor nulo para una clave o ruta JSON inexistente.

Sintaxis

```
JSON.MGET <key> [key ...] <path>
```

- clave (obligatorio): una o más claves Redis del tipo de documento.
- ruta (obligatoria): ruta JSON

Devolución

- Matriz de cadenas masivas. El tamaño de la matriz es igual al número de teclas del comando. Cada elemento de la matriz se rellena con (a) el comando JSON serializado tal como se encuentra en la ruta o (b) nulo si la clave no existe, la ruta no existe en el documento, o la ruta no es válida (error de sintaxis).
- Si alguna de las claves especificadas existe y no es una clave JSON, el comando devuelve el error `WRONGTYPE`.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '{"address":{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021"}}'
OK
127.0.0.1:6379> JSON.SET k2 . '{"address":{"street":"5 main
 Street","city":"Boston","state":"MA","zipcode":"02101"}}'
OK
127.0.0.1:6379> JSON.SET k3 . '{"address":{"street":"100 Park
 Ave","city":"Seattle","state":"WA","zipcode":"98102"}}'
OK
127.0.0.1:6379> JSON.MGET k1 k2 k3 $.address.city
1) "[\"New York\"]"
2) "[\"Boston\"]"
3) "[\"Seattle\"]"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"address":{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021"}}'
OK
127.0.0.1:6379> JSON.SET k2 . '{"address":{"street":"5 main
 Street","city":"Boston","state":"MA","zipcode":"02101"}}'
OK
127.0.0.1:6379> JSON.SET k3 . '{"address":{"street":"100 Park
 Ave","city":"Seattle","state":"WA","zipcode":"98102"}}'
OK

127.0.0.1:6379> JSON.MGET k1 k2 k3 .address.city
1) "\"New York\""
2) "\"Seattle\""
3) "\"Seattle\""
```

JSON.NUMINCRBY

Aumenta los valores numéricos de la ruta en un determinado número.

Sintaxis

```
JSON.NUMINCRBY <key> <path> <number>
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (obligatoria): ruta JSON
- número (obligatorio): un número

Devolución

Si la ruta es de sintaxis mejorada:

- Matriz de cadenas masivas que representa el valor resultante en cada ruta.
- Si un valor no es un número, su valor devuelto correspondiente es nulo.
- El error `WRONGTYPE` si el número no se puede analizar.
- El error `OVERFLOW` si el resultado está fuera del rango del doble IEEE de 64 bits.
- `NONEXISTENT` si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Cadena masiva que representa el valor resultante.
- Si se seleccionan varios valores, el comando devuelve el resultado del último valor actualizado.
- El error `WRONGTYPE` si el valor de la ruta no es un número.
- El error `WRONGTYPE` si el número no se puede analizar.
- El error `OVERFLOW` si el resultado está fuera del rango del doble IEEE de 64 bits.
- `NONEXISTENT` si la clave del documento no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 $.d[*] 10
"[11,12,13]"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[11,12,13]}"

127.0.0.1:6379> JSON.SET k1 $ '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 $.a[*] 1
```

```
"[]"
127.0.0.1:6379> JSON.NUMINCRBY k1 $.b[*] 1
"[2]"
127.0.0.1:6379> JSON.NUMINCRBY k1 $.c[*] 1
"[2,3]"
127.0.0.1:6379> JSON.NUMINCRBY k1 $.d[*] 1
"[2,3,4]"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,3],\"d\":[2,3,4]}"

127.0.0.1:6379> JSON.SET k2 $ '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k2 $.a.* 1
"[]"
127.0.0.1:6379> JSON.NUMINCRBY k2 $.b.* 1
"[2]"
127.0.0.1:6379> JSON.NUMINCRBY k2 $.c.* 1
"[2,3]"
127.0.0.1:6379> JSON.NUMINCRBY k2 $.d.* 1
"[2,3,4]"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":3},\"d\":{\"a\":2,\"b\":3,\"c\":4}}"

127.0.0.1:6379> JSON.SET k3 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k3 $.a.* 1
"[null]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.b.* 1
"[null,2]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.c.* 1
"[null,null]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.d.* 1
"[2,null,4]"
127.0.0.1:6379> JSON.GET k3
"{\"a\":{\"a\":\"a\"},\"b\":{\"a\":\"a\",\"b\":2},\"c\":{\"a\":\"a\",\"b\":\"b\"},\"d
\":{\"a\":2,\"b\":\"b\",\"c\":4}}"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
```

```
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 .d[1] 10
"12"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[1,12,3]}"

127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k1 .a[*] 1
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMINCRBY k1 .b[*] 1
"2"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[1,2],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMINCRBY k1 .c[*] 1
"3"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,3],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMINCRBY k1 .d[*] 1
"4"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,3],\"d\":[2,3,4]}"

127.0.0.1:6379> JSON.SET k2 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k2 .a.* 1
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMINCRBY k2 .b.* 1
"2"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":1,\"b\":2},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMINCRBY k2 .c.* 1
"3"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":3},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMINCRBY k2 .d.* 1
"4"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":3},\"d\":{\"a\":2,\"b\":3,\"c\":4}}"

127.0.0.1:6379> JSON.SET k3 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
```

```
127.0.0.1:6379> JSON.NUMINCRBY k3 .a.* 1
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMINCRBY k3 .b.* 1
"2"
127.0.0.1:6379> JSON.NUMINCRBY k3 .c.* 1
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMINCRBY k3 .d.* 1
"4"
```

JSON.NUMMULTBY

Multiplica los valores numéricos de la ruta por un determinado número.

Sintaxis

```
JSON.NUMMULTBY <key> <path> <number>
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (obligatoria): ruta JSON
- número (obligatorio): un número

Devolución

Si la ruta es de sintaxis mejorada:

- Matriz de cadenas masivas que representa el valor resultante en cada ruta.
- Si un valor no es un número, su valor devuelto correspondiente es nulo.
- El error `WRONGTYPE` si el número no se puede analizar.
- El error `OVERFLOW` si el resultado está fuera del rango del doble IEEE de 64 bits.
- `NONEXISTENT` si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Cadena masiva que representa el valor resultante.
- Si se seleccionan varios valores, el comando devuelve el resultado del último valor actualizado.
- El error `WRONGTYPE` si el valor de la ruta no es un número.

- El error `WRONGTYPE` si el número no se puede analizar.
- El error `OVERFLOW` si el resultado está fuera del rango del doble IEEE de 64 bits.
- `NONEXISTENT` si la clave del documento no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 $.d[*] 2
"[2,4,6]"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[2,4,6]}"

127.0.0.1:6379> JSON.SET k1 $ '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 $.a[*] 2
"[]"
127.0.0.1:6379> JSON.NUMMULTBY k1 $.b[*] 2
"[2]"
127.0.0.1:6379> JSON.NUMMULTBY k1 $.c[*] 2
"[2,4]"
127.0.0.1:6379> JSON.NUMMULTBY k1 $.d[*] 2
"[2,4,6]"

127.0.0.1:6379> JSON.SET k2 $ '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k2 $.a.* 2
"[]"
127.0.0.1:6379> JSON.NUMMULTBY k2 $.b.* 2
"[2]"
127.0.0.1:6379> JSON.NUMMULTBY k2 $.c.* 2
"[2,4]"
127.0.0.1:6379> JSON.NUMMULTBY k2 $.d.* 2
"[2,4,6]"

127.0.0.1:6379> JSON.SET k3 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k3 $.a.* 2
```

```
"[null]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.b.* 2
"[null,2]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.c.* 2
"[null,null]"
127.0.0.1:6379> JSON.NUMMULTBY k3 $.d.* 2
"[2,null,6]"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 .d[1] 2
"4"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[1],\"c\":[1,2],\"d\":[1,4,3]}"

127.0.0.1:6379> JSON.SET k1 . '{"a":[], "b":[1], "c":[1,2], "d":[1,2,3]}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k1 .a[*] 2
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMMULTBY k1 .b[*] 2
"2"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[1,2],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMMULTBY k1 .c[*] 2
"4"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,4],\"d\":[1,2,3]}"
127.0.0.1:6379> JSON.NUMMULTBY k1 .d[*] 2
"6"
127.0.0.1:6379> JSON.GET k1
"{\"a\":[],\"b\":[2],\"c\":[2,4],\"d\":[2,4,6]}"

127.0.0.1:6379> JSON.SET k2 . '{"a":{}, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1,
 "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k2 .a.* 2
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.NUMMULTBY k2 .b.* 2
"2"
127.0.0.1:6379> JSON.GET k2
```

```
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":1,\"b\":2},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMMULTBY k2 .c.* 2
"4"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":4},\"d\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.NUMMULTBY k2 .d.* 2
"6"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{},\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":4},\"d\":{\"a\":2,\"b\":4,\"c\":6}}"

127.0.0.1:6379> JSON.SET k3 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMMULTBY k3 .a.* 2
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMMULTBY k3 .b.* 2
"2"
127.0.0.1:6379> JSON.GET k3
"{\"a\":{\"a\":\"a\"},\"b\":{\"a\":\"a\",\"b\":2},\"c\":{\"a\":\"a\",\"b\":\"b\"},\"d
\":{\"a\":1,\"b\":\"b\",\"c\":3}}"
127.0.0.1:6379> JSON.NUMMULTBY k3 .c.* 2
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMMULTBY k3 .d.* 2
"6"
127.0.0.1:6379> JSON.GET k3
"{\"a\":{\"a\":\"a\"},\"b\":{\"a\":\"a\",\"b\":2},\"c\":{\"a\":\"a\",\"b\":\"b\"},\"d
\":{\"a\":2,\"b\":\"b\",\"c\":6}}"
```

JSON.OBJLEN

Obtiene el número de claves en los valores del objeto en la ruta.

Sintaxis

```
JSON.OBJLEN <key> [path]
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona

Devolución

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros, que representa la longitud del objeto en cada ruta.
- Si un valor no es un objeto, su valor devuelto correspondiente es nulo.
- Es nulo si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Entero, número de claves del objeto.
- Si hay varios objetos seleccionados, el comando devuelve la longitud del primer objeto.
- El error `WRONGTYPE` si el valor de la ruta no es un objeto.
- Error `WRONGTYPE` si la ruta no existe.
- Es nulo si la clave del documento no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJLEN k1 $.a
1) (integer) 0
127.0.0.1:6379> JSON.OBJLEN k1 $.a.*
(empty array)
127.0.0.1:6379> JSON.OBJLEN k1 $.b
1) (integer) 1
127.0.0.1:6379> JSON.OBJLEN k1 $.b.*
1) (nil)
127.0.0.1:6379> JSON.OBJLEN k1 $.c
1) (integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 $.c.*
1) (nil)
2) (nil)
127.0.0.1:6379> JSON.OBJLEN k1 $.d
1) (integer) 3
127.0.0.1:6379> JSON.OBJLEN k1 $.d.*
1) (nil)
2) (nil)
```

```
3) (integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 $.*
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 3
5) (nil)
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJLEN k1 .a
(integer) 0
127.0.0.1:6379> JSON.OBJLEN k1 .a.*
(error) NONEXISTENT JSON path does not exist
127.0.0.1:6379> JSON.OBJLEN k1 .b
(integer) 1
127.0.0.1:6379> JSON.OBJLEN k1 .b.*
(error) WRONGTYPE JSON element is not an object
127.0.0.1:6379> JSON.OBJLEN k1 .c
(integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 .c.*
(error) WRONGTYPE JSON element is not an object
127.0.0.1:6379> JSON.OBJLEN k1 .d
(integer) 3
127.0.0.1:6379> JSON.OBJLEN k1 .d.*
(integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 .*
(integer) 0
```

JSON.OBJKEYS

Obtiene los nombres de claves en los valores de objeto de la ruta.

Sintaxis

```
JSON.OBJKEYS <key> [path]
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona

Devolución

Si la ruta es de sintaxis mejorada:

- Matriz de matriz de cadenas masivas. Cada elemento es una matriz de claves de un objeto coincidente.
- Si un valor no es un objeto, su valor devuelto correspondiente es un valor vacío.
- Es nulo si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Matriz de cadenas a granel. Cada elemento es un nombre clave del objeto.
- Si hay varios objetos seleccionados, el comando devuelve las claves del primer objeto.
- El error `WRONGTYPE` si el valor de la ruta no es un objeto.
- Error `WRONGTYPE` si la ruta no existe.
- Es nulo si la clave del documento no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJKEYS k1 $.*
1) (empty array)
2) 1) "a"
3) 1) "a"
   2) "b"
4) 1) "a"
   2) "b"
   3) "c"
5) (empty array)
127.0.0.1:6379> JSON.OBJKEYS k1 $.d
1) 1) "a"
```

```
   2) "b"
   3) "c"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{}, "b":{"a":"a"}, "c":{"a":"a", "b":"bb"}, "d":
{"a":1, "b":"b", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJKEYS k1 .*
1) "a"
127.0.0.1:6379> JSON.OBJKEYS k1 .d
1) "a"
2) "b"
3) "c"
```

JSON.RESP

Devuelve el valor JSON en la ruta dada en Redis Serialization Protocol (RESP). Si el valor es contenedor, la respuesta es una matriz RESP o matriz anidada.

- El valor nulo de JSON se asigna a la cadena masiva nula de RESP.
- Los valores booleanos JSON se asignan a las cadenas simples de RESP respectivas.
- Los números enteros se asignan a números enteros RESP.
- Los números de coma flotante doble IEEE de 64 bits se asignan a cadenas masivas RESP.
- Las cadenas JSON se asignan a cadenas masivas de RESP.
- Las matrices JSON se representan como matrices RESP, donde el primer elemento es la cadena simple [, seguida de los elementos de la matriz.
- Los objetos JSON se representan como matrices RESP, donde el primer elemento es la cadena simple {, seguida de los pares clave-valor, cada uno de los cuales es una cadena masiva RESP.

Sintaxis

```
JSON.RESP <key> [path]
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona

Devolución

Si la ruta es de sintaxis mejorada:

- Matriz de matrices. Cada elemento de la matriz representa la forma RESP del valor en una ruta.
- Matriz vacía si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Matriz, que representa la forma RESP del valor en la ruta.
- Es nulo si la clave del documento no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK

127.0.0.1:6379> JSON.RESP k1 $.address
1) 1) {
   2) 1) "street"
      2) "21 2nd Street"
   3) 1) "city"
      2) "New York"
   4) 1) "state"
      2) "NY"
   5) 1) "zipcode"
      2) "10021-3100"

127.0.0.1:6379> JSON.RESP k1 $.address.*
1) "21 2nd Street"
2) "New York"
3) "NY"
4) "10021-3100"
```

```
127.0.0.1:6379> JSON.RESP k1 $.phoneNumbers
1) 1) [
   2) 1) {
      2) 1) "type"
         2) "home"
      3) 1) "number"
         2) "555 555-1234"
   3) 1) {
      2) 1) "type"
         2) "office"
      3) 1) "number"
         2) "555 555-4567"

127.0.0.1:6379> JSON.RESP k1 $.phoneNumbers[*]
1) 1) {
   2) 1) "type"
      2) "home"
   3) 1) "number"
      2) "212 555-1234"
2) 1) {
   2) 1) "type"
      2) "office"
   3) 1) "number"
      2) "555 555-4567"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK

127.0.0.1:6379> JSON.RESP k1 .address
1) {
2) 1) "street"
   2) "21 2nd Street"
3) 1) "city"
   2) "New York"
4) 1) "state"
```

```
   2) "NY"
5) 1) "zipcode"
   2) "10021-3100"

127.0.0.1:6379> JSON.RESP k1
 1) {
 2) 1) "firstName"
    2) "John"
 3) 1) "lastName"
    2) "Smith"
 4) 1) "age"
    2) (integer) 27
 5) 1) "weight"
    2) "135.25"
 6) 1) "isAlive"
    2) true
 7) 1) "address"
    2) 1) {
       2) 1) "street"
          2) "21 2nd Street"
       3) 1) "city"
          2) "New York"
       4) 1) "state"
          2) "NY"
       5) 1) "zipcode"
          2) "10021-3100"
 8) 1) "phoneNumbers"
    2) 1) [
       2) 1) {
          2) 1) "type"
             2) "home"
          3) 1) "number"
             2) "212 555-1234"
       3) 1) {
          2) 1) "type"
             2) "office"
          3) 1) "number"
             2) "555 555-4567"
 9) 1) "children"
    2) 1) [
10) 1) "spouse"
    2) (nil)
```

JSON.SET

Establece valores JSON en la ruta.

Si la ruta de acceso llama a un miembro de objeto:

- Si el elemento principal no existe, el comando devolverá un error INEXISTENTE.
- Si el elemento principal existe pero no es un objeto, el comando devolverá ERROR.
- Si el elemento principal existe y es un objeto:
  - Si el miembro no existe, se anexará un miembro nuevo al objeto principal si y solo si el objeto principal es el último objeto secundario de la ruta. De lo contrario, el comando devolverá un error INEXISTENTE.
  - Si el miembro existe, su valor se reemplazará por el valor JSON.

Si la ruta requiere un índice de matriz:

- Si el elemento principal no existe, el comando devolverá un error INEXISTENTE.
- Si el elemento principal existe pero no es una matriz, el comando devolverá ERROR.
- Si el elemento principal existe pero el índice está fuera de los límites, el comando devuelve un error OUTOFBOUNDARIES.
- Si el elemento principal existe y el índice es válido, el elemento se reemplazará por el nuevo valor JSON.

Si la ruta llama a un objeto o matriz, el valor (objeto o matriz) se reemplazará por el nuevo valor JSON.

Sintaxis

```
JSON.SET <key> <path> <json> [NX | XX]
```

[NX | XX] Donde puede tener 0 o 1 de [NX | XX] identificadores

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (obligatoria): una ruta JSON. Para una nueva clave de Redis, la ruta JSON debe ser la raíz “.”.
- NX (opcional): si la ruta es la raíz, establezca el valor solo si la clave de Redis no existe, por ejemplo, insertar un nuevo documento. Si la ruta no es la raíz, establece el valor solo si la ruta no existe, es decir, inserta un valor en el documento.

- XX (opcional): si la ruta es la raíz, establezca el valor solo si existe la clave de Redis, por ejemplo, reemplazar el documento existente. Si la ruta no es la raíz, establece el valor solo si la ruta existe, es decir, actualiza el valor existente.

Devolución

- Cadena simple ‘OK’ en caso de éxito.
- Es nulo si no se cumple la condición NX o XX.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{"a":1, "b":2, "c":3}}'
OK
127.0.0.1:6379> JSON.SET k1 $.a.* '0'
OK
127.0.0.1:6379> JSON.GET k1
"{\"a\":{\"a\":0,\"b\":0,\"c\":0}}"

127.0.0.1:6379> JSON.SET k2 . '{"a": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.SET k2 $.a[*] '0'
OK
127.0.0.1:6379> JSON.GET k2
"{\"a\":[0,0,0,0,0]}"
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"c":{"a":1, "b":2}, "e": [1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.SET k1 .c.a '0'
OK
127.0.0.1:6379> JSON.GET k1
"{\"c\":{\"a\":0,\"b\":2},\"e\":[1,2,3,4,5]}"
127.0.0.1:6379> JSON.SET k1 .e[-1] '0'
OK
127.0.0.1:6379> JSON.GET k1
"{\"c\":{\"a\":0,\"b\":2},\"e\":[1,2,3,4,0]}"
127.0.0.1:6379> JSON.SET k1 .e[5] '0'
```

```
(error) OUTOFBOUNDARIES Array index is out of bounds
```

JSON.STRAPPEND

Adjunta una cadena a las cadenas JSON de la ruta.

Sintaxis

```
JSON.STRAPPEND <key> [path] <json_string>
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona
- json_string (obligatorio): representación JSON de una cadena. Tenga en cuenta que se debe citar una cadena JSON, por ejemplo, “foo”.

Devolución

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros, que representa la nueva longitud de la cadena en cada ruta.
- Si un valor en la ruta no es una cadena, su valor devuelto correspondiente es nulo.
- Error `SYNTAXERR` si el argumento json de entrada no es una cadena JSON válida.
- Error `NONEXISTENT` si la ruta no existe.

Si la ruta es de sintaxis restringida:

- Entero, la nueva longitud de la cadena.
- Si se seleccionan varios valores de cadena, el comando devuelve la nueva longitud de la última cadena actualizada.
- Error `WRONGTYPE` si el valor de la ruta no es una cadena.
- Error `WRONGTYPE` si el argumento json de entrada no es una cadena JSON válida.
- Error `NONEXISTENT` si la ruta no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRAPPEND k1 $.a.a '"a"'
1) (integer) 2
127.0.0.1:6379> JSON.STRAPPEND k1 $.a.* '"a"'
1) (integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 $.b.* '"a"'
1) (integer) 2
2) (nil)
127.0.0.1:6379> JSON.STRAPPEND k1 $.c.* '"a"'
1) (integer) 2
2) (integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 $.c.b '"a"'
1) (integer) 4
127.0.0.1:6379> JSON.STRAPPEND k1 $.d.* '"a"'
1) (nil)
2) (integer) 2
3) (nil)
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRAPPEND k1 .a.a '"a"'
(integer) 2
127.0.0.1:6379> JSON.STRAPPEND k1 .a.* '"a"'
(integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 .b.* '"a"'
(integer) 2
127.0.0.1:6379> JSON.STRAPPEND k1 .c.* '"a"'
(integer) 3
127.0.0.1:6379> JSON.STRAPPEND k1 .c.b '"a"'
(integer) 4
127.0.0.1:6379> JSON.STRAPPEND k1 .d.* '"a"'
(integer) 2
```

JSON.STRLEN

Obtiene las longitudes de los valores de cadena JSON en la ruta.

Sintaxis

```
JSON.STRLEN <key> [path]
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona

Devolución

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros, que representa la longitud de la cadena en cada ruta.
- Si un valor no es una cadena, su valor devuelto correspondiente es nulo.
- Es nulo si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Entero, la longitud de la cadena.
- Si se seleccionan varios valores de cadena, el comando devuelve la longitud de la primera cadena.
- Error `WRONGTYPE` si el valor de la ruta no es una cadena.
- Error `NONEXISTENT` si la ruta no existe.
- Es nulo si la clave del documento no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRLEN k1 $.a.a
1) (integer) 1
127.0.0.1:6379> JSON.STRLEN k1 $.a.*
```

```
1) (integer) 1
127.0.0.1:6379> JSON.STRLEN k1 $.c.*
1) (integer) 1
2) (integer) 2
127.0.0.1:6379> JSON.STRLEN k1 $.c.b
1) (integer) 2
127.0.0.1:6379> JSON.STRLEN k1 $.d.*
1) (nil)
2) (integer) 1
3) (nil)
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 $ '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"bb"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.STRLEN k1 .a.a
(integer) 1
127.0.0.1:6379> JSON.STRLEN k1 .a.*
(integer) 1
127.0.0.1:6379> JSON.STRLEN k1 .c.*
(integer) 1
127.0.0.1:6379> JSON.STRLEN k1 .c.b
(integer) 2
127.0.0.1:6379> JSON.STRLEN k1 .d.*
(integer) 1
```

JSON.TOGGLE

Alterna los valores booleanos entre verdadero y falso en la ruta.

Sintaxis

```
JSON.TOGGLE <key> [path]
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona

Devolución

Si la ruta es de sintaxis mejorada:

- Matriz de números enteros (0 - falso, 1 - verdadero) que representa el valor booleano resultante en cada ruta.
- Si un valor no es un valor booleano, su valor devuelto correspondiente es nulo.
- `NONEXISTENT` si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Cadena (“verdadero”/”falso”) que representa el valor booleano resultante.
- `NONEXISTENT` si la clave del documento no existe.
- Error `WRONGTYPE` si el valor de la ruta no es un valor booleano.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '{"a":true, "b":false, "c":1, "d":null, "e":"foo", "f":
[], "g":{}}'
OK
127.0.0.1:6379> JSON.TOGGLE k1 $.*
1) (integer) 0
2) (integer) 1
3) (nil)
4) (nil)
5) (nil)
6) (nil)
7) (nil)
127.0.0.1:6379> JSON.TOGGLE k1 $.*
1) (integer) 1
2) (integer) 0
3) (nil)
4) (nil)
5) (nil)
6) (nil)
7) (nil)
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 . true
OK
127.0.0.1:6379> JSON.TOGGLE k1
"false"
127.0.0.1:6379> JSON.TOGGLE k1
"true"

127.0.0.1:6379> JSON.SET k2 . '{"isAvailable": false}'
OK
127.0.0.1:6379> JSON.TOGGLE k2 .isAvailable
"true"
127.0.0.1:6379> JSON.TOGGLE k2 .isAvailable
"false"
```

JSON.TYPE

Informa el tipo de valores en la ruta dada.

Sintaxis

```
JSON.TYPE <key> [path]
```

- clave (obligatorio): clave Redis del tipo de documento JSON
- ruta (opcional): una ruta JSON. Toma el valor predeterminado raíz si no se proporciona

Devolución

Si la ruta es de sintaxis mejorada:

- Matriz de cadenas, que representa el tipo de valor en cada ruta. El tipo es uno de {"nulo", "booleano", "cadena", "número", "entero", "objeto" y "matriz"}.
- Si no existe una ruta, su valor de retorno correspondiente es nulo.
- Matriz vacía si la clave del documento no existe.

Si la ruta es de sintaxis restringida:

- Cadena, tipo del valor
- Es nulo si la clave del documento no existe.

- Es nulo si la ruta JSON no es válida o no existe.

Ejemplos

Sintaxis de la ruta mejorada:

```
127.0.0.1:6379> JSON.SET k1 . '[1, 2.3, "foo", true, null, {}, []]'
OK
127.0.0.1:6379> JSON.TYPE k1 $[*]
1) integer
2) number
3) string
4) boolean
5) null
6) object
7) array
```

Sintaxis de la ruta restringida:

```
127.0.0.1:6379> JSON.SET k1 .
 '{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
{"street":"21 2nd Street","city":"New
 York","state":"NY","zipcode":"10021-3100"},"phoneNumbers":
[{"type":"home","number":"212 555-1234"},{"type":"office","number":"646
 555-4567"}],"children":[],"spouse":null}'
OK
127.0.0.1:6379> JSON.TYPE k1
object
127.0.0.1:6379> JSON.TYPE k1 .children
array
127.0.0.1:6379> JSON.TYPE k1 .firstName
string
127.0.0.1:6379> JSON.TYPE k1 .age
integer
127.0.0.1:6379> JSON.TYPE k1 .weight
number
127.0.0.1:6379> JSON.TYPE k1 .isAlive
boolean
127.0.0.1:6379> JSON.TYPE k1 .spouse
null
```

Etiquetado de los recursos de MemoryDB

Para ayudarlo a administrar sus clústeres y otros recursos de MemoryDB, puede asignar sus propios metadatos a cada recurso en forma de etiquetas. Las etiquetas le permiten clasificar los recursos de AWS de diversas maneras, por ejemplo, según su finalidad, propietario o entorno. Esto es útil cuando tiene muchos recursos del mismo tipo: puede identificar rápidamente un recurso específico en función de las etiquetas que le haya asignado. En este tema se describe qué son las etiquetas y cómo crearlas.

> **Warning**
>
> Como práctica recomendada, no debe incluir datos confidenciales en las etiquetas.

Conceptos básicos de etiquetas

Una etiqueta es una marca que se asigna a un recurso de AWS. Cada etiqueta está formada por una clave y un valor opcional, ambos definidos por el usuario. Las etiquetas permiten clasificar los recursos de AWS de diversas maneras, por ejemplo, según finalidad o propietario. Por ejemplo, podría definir un conjunto de etiquetas para los clústeres de MemoryDB de su cuenta que lo ayude a realizar un seguimiento del propietario y el grupo de usuarios de cada clúster.

Recomendamos que idee un conjunto de claves de etiqueta que cumpla sus necesidades para cada tipo de recurso. Mediante el uso de un conjunto coherente de claves de etiquetas, podrá administrar los recursos más fácilmente. Puede buscar y filtrar los recursos en función de las etiquetas que agregue. Para obtener más información acerca de cómo implementar una estrategia eficaz de etiquetado de recursos, consulte el documento técnico de Prácticas recomendadas de etiquetado de AWS.

Las etiquetas no tienen ningún significado semántico para MemoryDB y se interpretan estrictamente como cadenas de caracteres. Además, las etiquetas no se asignan a los recursos automáticamente. Puede editar las claves y los valores de las etiquetas y también puede eliminar etiquetas de un recurso en cualquier momento. Puede establecer el valor de una etiqueta en `null`. Si añade una etiqueta con la misma clave que una etiqueta existente en ese recurso, el nuevo valor sobrescribirá al antiguo. Si elimina un recurso, también se eliminará cualquier etiqueta asignada a dicho recurso.

Puede trabajar con etiquetas utilizando la AWS Management Console, la AWS CLI y la API de MemoryDB.

Si utiliza IAM, puede controlar qué usuarios de su cuenta de AWS tienen permiso para crear, editar o eliminar etiquetas. Para obtener más información, consulte Permisos de nivel de recursos.

Recursos que se pueden etiquetar

Puede etiquetar la mayoría de los recursos de MemoryDB que ya existen en la cuenta. La siguiente tabla enumera los recursos que admiten etiquetas. Si utiliza la AWS Management Console, puede aplicar etiquetas a recursos a través del Editor de etiquetas. Algunas pantallas de recursos permiten especificar etiquetas para un recurso al crear dicho recurso; por ejemplo, una etiqueta con una clave de Name (Nombre) y un valor que especifique. En la mayoría de los casos, la consola aplica las etiquetas inmediatamente después de crear el recurso (y no durante la creación del mismo). La consola puede organizar los recursos según la etiqueta Nombre, si bien dicha etiqueta no tiene significado semántico para el servicio de MemoryDB.

Además, algunas acciones de creación de recursos le permiten especificar etiquetas para un recurso al crear dicho recurso. Si no se pueden aplicar etiquetas durante la creación del recurso, el proceso de creación del recurso se revierte. Esto garantiza que los recursos se creen con etiquetas o, de lo contrario, no se creen y que ningún recurso se quede jamás sin etiquetar. Al etiquetar los recursos en el momento de su creación, se eliminar la necesidad de ejecutar scripts de etiquetado personalizados tras la creación del recurso.

Si utiliza la API de Amazon MemoryDB, la CLI de AWS o un SDK de AWS, puede aplicar etiquetas mediante el parámetro `Tags` en la acción de la API de MemoryDB pertinente. Son:

- `CreateCluster`
- `CopySnapshot`
- `CreateParameterGroup`
- `CreateSubnetGroup`
- `CreateSnapshot`
- `CreateACL`
- `CreateUser`

En la tabla siguiente se describen los recursos de MemoryDB que se pueden etiquetar y aquellos que se pueden etiquetar en el momento de su creación con la API de MemoryDB, la CLI de AWS o un SDK de AWS.

Compatibilidad con el etiquetado de recursos de MemoryDB

Admite etiquetas	Admite el etiquetad o durante la creación
Sí	Sí
Sí	Sí
Sí	Sí
Sí	Sí
Sí	Sí
Sí	Sí

En las políticas de IAM, puede aplicar permisos de nivel de recursos basados en etiquetas a las acciones de la API de MemoryDB que admitan el etiquetado durante la creación para implementar un control pormenorizado de los usuarios y los grupos que pueden etiquetar recursos durante la creación. Sus recursos se encuentran debidamente protegidos de las etiquetas de creación que se aplican de inmediato a los recursos. Por lo tanto, cualquier permiso de nivel de recursos basado en etiquetas que controle la utilización de recursos es efectivo de inmediato. Se puede realizar un seguimiento y un registro más precisos de los recursos. Puede establecer el etiquetado obligatorio de los nuevos recursos y controlar qué claves y valores de etiquetas se usan en ellos.

Para obtener más información, consulte Ejemplos de etiquetado de recursos.

A fin de obtener más información sobre el etiquetado de recursos para facturación, consulte Monitoreo de costos con etiquetas de asignación de costos.

Etiquetado de clústeres e instantáneas

Las siguientes reglas se aplican al etiquetado como parte de las operaciones de solicitud:

- CreateCluster:
  - Si se proporciona el `--cluster-name`:

Si se incluyen etiquetas en la solicitud, solo se etiquetará el clúster.

- Si se proporciona el `--snapshot-name`:

  Si se incluyen etiquetas en la solicitud, solo se le aplicarán esas etiquetas al clúster. Si no se incluyen etiquetas en la solicitud, las etiquetas de la instantánea se agregarán al clúster.

- CreateSnapshot
  - Si se proporciona el `--cluster-name`:

    Si no se incluyen etiquetas en la solicitud, las etiquetas de solicitud se agregarán a la instantánea. Si no se incluyen etiquetas en la solicitud, las etiquetas del clúster se agregarán a la instantánea.

  - Para las instantáneas automáticas:

    Las etiquetas se propagarán desde las etiquetas de agrupamiento.

- CopySnapshot

  Si no se incluyen etiquetas en la solicitud, las etiquetas de solicitud se agregarán a la instantánea. Si no se incluyen etiquetas en la solicitud, las etiquetas de la instantánea fuente se agregarán a la instantánea copiada.

- TagResource y UntagResource:

  Las etiquetas se añadirán o eliminarán del recurso.

Restricciones de las etiquetas

Se aplican las siguientes restricciones básicas a las etiquetas:

- Número máximo de etiquetas por recurso: 50
- Para cada recurso, cada clave de etiqueta debe ser única y solo puede tener un valor.
- Longitud máxima de la clave: 128 caracteres Unicode en UTF-8.
- Longitud máxima del valor: 256 caracteres Unicode en UTF-8.
- Si bien MemoryDB admite utilizar cualquier carácter en sus etiquetas, otros servicios pueden ser restrictivos. Los caracteres permitidos en los servicios son: letras, números y espacios representables en UTF-8, además de los siguientes caracteres: + - = . _ : / @
- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.

- El prefijo `aws:` se reserva para uso de AWS. Si la etiqueta tiene una clave de etiqueta con este prefijo, no puede editar ni eliminar la clave o el valor de la etiqueta. Las etiquetas que tengan el prefijo `aws:` no cuentan para el límite de etiquetas por recurso.

No puede terminar, detener ni eliminar un recurso basado únicamente en sus etiquetas; debe especificar el identificador del recurso. Por ejemplo, para eliminar instantáneas que etiquetó con una clave de etiqueta llamada `DeleteMe`, debe utilizar la acción `DeleteSnapshot` con los identificadores del recurso de las instantáneas, como `snap-1234567890abcdef0`.

Para obtener más información sobre los recursos de MemoryDB que puede etiquetar, consulte Recursos que se pueden etiquetar.

Ejemplos de etiquetado de recursos

- Agregar etiquetas a un clúster.

```
aws memorydb tag-resource \
--resource-arn arn:aws:memorydb:us-east-1:111111222233:cluster/my-cluster \
--tags Key="project",Value="XYZ" Key="memorydb",Value="Service"
```

- Creación de un clúster mediante etiquetas.

```
aws memorydb create-cluster \
--cluster-name testing-tags \
--description cluster-test \
--subnet-group-name test \
--node-type db.r6g.large \
--acl-name open-access \
--tags Key="project",Value="XYZ" Key="memorydb",Value="Service"
```

- Creación de una instantánea con etiquetas.

  En este caso, si agrega etiquetas a la solicitud, incluso si el clúster contiene etiquetas, la instantánea solo recibirá las etiquetas de la solicitud.

```
aws memorydb create-snapshot \
--cluster-name testing-tags \
--snapshot-name bkp-testing-tags-mycluster \
--tags Key="work",Value="foo"
```

Monitoreo de costos con etiquetas de asignación de costos

Al agregar etiquetas de asignación de costos a sus recursos en MemoryDB para Redis, puede realizar un seguimiento de los costos agrupando los gastos en sus facturas por valores de etiqueta de recursos.

Las etiquetas de asignación de costos de MemoryDB son pares clave-valor que el usuario define y asocia a un recurso de MemoryDB. Las claves y los valores distinguen entre mayúsculas y minúsculas. Puede utilizar una clave de etiqueta para definir una categoría y el valor de la etiqueta puede ser un elemento dentro de esa categoría. Por ejemplo, puede definir una clave de etiqueta `CostCenter` y un valor de etiqueta `10010` para indicar que el recurso va asignado al centro de costos 10010. También puede usar etiquetas para designar recursos para pruebas o para producción a través de una clave como `Environment` y valores como `test` o `production`. Se recomienda utilizar un conjunto coherente de claves de etiqueta que facilite el seguimiento de los costos asociados a los recursos.

Utilice las etiquetas de asignación de costos para organizar la factura de AWS de modo que refleje su propia estructura de costos. Para ello, inscríbase para obtener una factura de la cuenta de AWS que incluya valores de clave de etiquetas. A continuación, para ver los costos de los recursos combinados, organice la información de facturación de acuerdo con los recursos con los mismos valores de clave de etiquetas. Por ejemplo, puede etiquetar varios recursos con un nombre de aplicación específico y luego organizar su información de facturación para ver el costo total de la aplicación en distintos servicios.

También puede combinar etiquetas para realizar un seguimiento de los costos con un mayor nivel de detalle. Por ejemplo, para realizar un seguimiento de los costos de su servicio por región, puede utilizar las claves de etiqueta `Service` y `Region`. En un recurso podría tener los valores `MemoryDB` y `Asia Pacific (Singapore)` y en otro recurso, los valores `MemoryDB` y `Europe (Frankfurt)`. A continuación, puede ver el total de costos de MemoryDB desglosados por región. Para obtener más información, consulte Uso de etiquetas de asignación de costos en la Guía del usuario de AWS Billing.

Puede agregar etiquetas de asignación de costos de MemoryDB a los clústeres de MemoryDB. Al agregar, enumerar, modificar, copiar o quitar una etiqueta, la operación se aplica únicamente al clúster especificado.

Características de las etiquetas de asignación de costos de MemoryDB

- Las etiquetas de asignación de costos se aplican a recursos de MemoryDB especificados en operaciones de la API y de la CLI como ARN. El tipo de recurso será "clúster".

  Formato de ARN: `arn:aws:memorydb:`*`<region>`*`:`*`<customer-id>`*`:`*`<resource-type>`*`/`*`<resource-name>`*

  Ejemplo de ARN: `arn:aws:memorydb:us-east-1:1234567890:cluster/my-cluster`
- La clave de la etiqueta es el nombre obligatorio de la etiqueta. El valor de cadena de la clave puede tener una longitud de entre 1 y 128 caracteres Unicode y no puede llevar el prefijo `aws:`. La cadena solo puede contener un conjunto Unicode de letras, dígitos, espacios en blanco, guiones bajos (_), puntos (.), dos puntos (:), barras oblicuas (\ ), signos de igual (=), signos de suma (+), guiones (-) o signos de arroba (@).
- El valor de etiqueta es la parte opcional de la etiqueta. El valor de cadena del valor puede tener una longitud de entre 1 y 256 caracteres Unicode y no puede llevar el prefijo `aws:`. La cadena solo puede contener un conjunto Unicode de letras, dígitos, espacios en blanco, guiones bajos (_), puntos (.), dos puntos (:), barras oblicuas (\ ), signos de igual (=), signos de suma (+), guiones (-) o signos de arroba (@).
- Un recurso de MemoryDB puede tener un máximo de 50 etiquetas.
- Los valores no deben ser únicos dentro de un conjunto de etiquetas. Por ejemplo, puede disponer de un conjunto de etiquetas donde las claves `Service` y `Application` tienen el valor `MemoryDB`.

AWS no aplica ningún significado semántico a las etiquetas. Las etiquetas se interpretan estrictamente como cadenas de caracteres. AWS no establece automáticamente ninguna etiqueta en ningún recurso de MemoryDB.

Administración de etiquetas de asignación de costos mediante la AWS CLI

Puede utilizar la AWS CLI para agregar, modificar o quitar etiquetas de asignación de costos.

Ejemplo de ARN: `arn:aws:memorydb:us-east-1:1234567890:cluster/my-cluster`

Temas

- Enumeración de etiquetas mediante la AWS CLI
- Adición de etiquetas mediante la AWS CLI
- Modificación de etiquetas mediante la AWS CLI

- Eliminación de etiquetas mediante la AWS CLI

Enumeración de etiquetas mediante la AWS CLI

Puede utilizar la AWS CLI para obtener una lista de las etiquetas de un recurso de MemoryDB existente utilizando la operación list-tags.

El código siguiente utiliza la AWS CLI para obtener una lista de las etiquetas del clúster de MemoryDB `my-cluster` de la región us-east-1.

Para Linux, macOS o Unix:

```
aws memorydb list-tags \
  --resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster
```

Para Windows:

```
aws memorydb list-tags ^
  --resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster
```

La salida de esta operación se parecerá a lo siguiente, una lista de todas las etiquetas en el recurso.

```
{
   "TagList": [
      {
         "Value": "10110",
         "Key": "CostCenter"
      },
      {
         "Value": "EC2",
         "Key": "Service"
      }
   ]
}
```

Si no hay etiquetas en el recurso, la salida será una TagList vacía.

```
{
   "TagList": []
}
```

Para obtener más información, consulte list-tags de la AWS CLI de MemoryDB.

Adición de etiquetas mediante la AWS CLI

Puede utilizar la AWS CLI para agregar etiquetas a un recurso de MemoryDB existente utilizando la operación de la CLI tag-resource. Si la clave de etiqueta no existe en el recurso, la clave y el valor se añadirán a los recursos. Si la clave ya existe en el recurso, el valor asociado a dicha clave se actualizará al nuevo valor.

El código siguiente utiliza la AWS CLI para agregar las claves `Service` y `Region` con los valores `memorydb` y `us-east-1` respectivamente al clúster `my-cluster` de la región us-east-1.

Para Linux, macOS o Unix:

```
aws memorydb tag-resource \
 --resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster \
 --tags Key=Service,Value=memorydb \
        Key=Region,Value=us-east-1
```

Para Windows:

```
aws memorydb tag-resource ^
 --resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster ^
 --tags Key=Service,Value=memorydb ^
        Key=Region,Value=us-east-1
```

Tras la operación, la salida de esta operación se parecerá a lo siguiente, una lista de todas las etiquetas en el recurso.

```
{
   "TagList": [
      {
         "Value": "memorydb",
         "Key": "Service"
      },
      {
         "Value": "us-east-1",
         "Key": "Region"
      }
   ]
}
```

Para obtener más información, consulte la AWS CLI para MemoryDB tag-resource.

También puede utilizar la AWS CLI para agregar etiquetas a un clúster al crear un nuevo clúster utilizando la operación create-cluster.

Modificación de etiquetas mediante la AWS CLI

Puede utilizar la AWS CLI para modificar las etiquetas de un clúster de MemoryDB.

Para modificar las etiquetas:

- Use tag-resource para agregar una etiqueta y un valor nuevos o para cambiar el valor asociado a una etiqueta existente.
- Use untag-resource para quitar etiquetas especificadas del recurso.

La salida de cualquier operación será una lista de las etiquetas y sus valores en el clúster especificado.

Eliminación de etiquetas mediante la AWS CLI

Puede utilizar la AWS CLI para quitar etiquetas de un clúster de MemoryDB existente utilizando la operación untag-resource.

El código siguiente utiliza la AWS CLI para quitar las etiquetas con las claves `Service` y `Region` del clúster `my-cluster` de la región us-east-1.

Para Linux, macOS o Unix:

```
aws memorydb untag-resource \
 --resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster \
 --tag-keys Region Service
```

Para Windows:

```
aws memorydb untag-resource ^
 --resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster ^
 --tag-keys Region Service
```

Tras la operación, la salida de esta operación se parecerá a lo siguiente, una lista de todas las etiquetas en el recurso.

```
{
   "TagList": []
}
```

Para obtener más información, consulte untag-resource de la AWS CLI para MemoryDB.

Administración de etiquetas de asignación de costos mediante la API de MemoryDB

Puede utilizar la API de MemoryDB para agregar, modificar o quitar etiquetas de asignación de costos.

Las etiquetas de asignación de costos se aplican a los clústeres de MemoryDB. El clúster que se va a etiquetar se especifica mediante un ARN (nombre de recurso de Amazon).

Ejemplo de ARN: `arn:aws:memorydb:us-east-1:1234567890:cluster/my-cluster`

Temas

- Enumeración de etiquetas mediante la API de MemoryDB
- Adición de etiquetas mediante la API de MemoryDB
- Modificación de etiquetas con la API de MemoryDB
- Eliminación de etiquetas mediante la API de MemoryDB

Enumeración de etiquetas mediante la API de MemoryDB

Puede utilizar la API de MemoryDB para obtener una lista de las etiquetas de un recurso existente utilizando la operación ListTags.

El código siguiente utiliza la API de MemoryDB para obtener una lista de las etiquetas del recurso `my-cluster` de la región us-east-1.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=ListTags
   &ResourceArn=arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Version=2021-01-01
```

```
   &Timestamp=20210802T192317Z
   &X-Amz-Credential=<credential>
```

Adición de etiquetas mediante la API de MemoryDB

Puede utilizar la API de MemoryDB para agregar etiquetas a un clúster de MemoryDB existente utilizando la operación TagResource. Si la clave de etiqueta no existe en el recurso, la clave y el valor se añadirán a los recursos. Si la clave ya existe en el recurso, el valor asociado a dicha clave se actualizará al nuevo valor.

El código siguiente utiliza la API de MemoryDB para añadir las claves `Service` y `Region` con los valores `memorydb` y `us-east-1` respectivamente al recurso `my-cluster` en la región us-east-1.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=TagResource
   &ResourceArn=arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Tags.member.1.Key=Service
   &Tags.member.1.Value=memorydb
   &Tags.member.2.Key=Region
   &Tags.member.2.Value=us-east-1
   &Version=2021-01-01
   &Timestamp=20210802T192317Z
   &X-Amz-Credential=<credential>
```

Para obtener más información, consulte TagResource.

Modificación de etiquetas con la API de MemoryDB

Puede utilizar la API de MemoryDB para modificar las etiquetas de un clúster de MemoryDB.

Para modificar el valor de una etiqueta:

- Use la operación TagResource para agregar una etiqueta y un valor nuevos o para cambiar el valor de una etiqueta existente.
- Para quitar etiquetas de un recurso, utilice la acción UntagResource.

La salida de cualquier operación será una lista de las etiquetas y sus valores en el recurso especificado.

Eliminación de etiquetas mediante la API de MemoryDB

Puede utilizar la API de MemoryDB para quitar etiquetas de un clúster de MemoryDB existente utilizando la operación UntagResource.

El código siguiente utiliza la API de MemoryDB para quitar las etiquetas con las claves `Service` y `Region` del clúster `my-cluster` de la región us-east-1.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=UntagResource
   &ResourceArn=arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &TagKeys.member.1=Service
   &TagKeys.member.2=Region
   &Version=2021-01-01
   &Timestamp=20210802T192317Z
   &X-Amz-Credential=<credential>
```

Administración del mantenimiento

Cada clúster tiene un periodo de mantenimiento semanal durante el que se aplican los cambios del sistema. Si no especifica un periodo de mantenimiento preferido al crear o modificar un clúster, MemoryDB asignará un periodo de mantenimiento de 60 minutos en el periodo de mantenimiento de su región de un día de la semana elegido al azar.

El periodo de mantenimiento de 60 minutos se elige al azar de un bloque de 8 horas por cada región. En la siguiente tabla, se muestran los bloques de tiempo de cada región desde los que se asignan los periodos predeterminados de mantenimiento. Puede elegir un periodo de mantenimiento preferido fuera del bloque del periodo de mantenimiento de la región.

Código de región	Nombre de la región de	Periodo de mantenimiento de la región
`ap-northeast-1`	Región Asia-Pacífico (Tokio)	13:00 — 21:00 UTC
`ap-northeast-2`	Región Asia-Pacífico (Seúl)	12:00 — 20:00 UTC
`ap-south-1`	Región Asia-Pacífico (Mumbai)	17:30 — 01:30 UTC

Código de región	Nombre de la región de	Periodo de mantenimiento de la región
ap-southeast-1	Región Asia-Pacífico (Singapur)	14:00 — 22:00 UTC
ap-east-1	Región Asia Pacífico (Hong Kong)	13:00 — 21:00 UTC
ap-southeast-2	Región Asia-Pacífico (Sídney)	12:00 — 20:00 UTC
cn-north-1	Región China (Pekín)	14:00 — 22:00 UTC
cn-northwest-1	Región China (Ningxia)	14:00 — 22:00 UTC
eu-west-3	Región EU (París)	23:59 — 07:29 UTC
eu-central-1	Europe (Frankfurt) Region	23:00 — 07:00 UTC
eu-west-1	Europe (Ireland) Region	22:00 — 06:00 UTC
eu-west-2	Europe (London) Region	23:00 — 07:00 UTC
sa-east-1	South America (São Paulo) Region	01:00 — 09:00 UTC
ca-central-1	Canada (Central) Region	03:00 — 11:00 UTC
us-east-1	Región del este de EE. UU (N. Virginia)	03:00 — 11:00 UTC
us-east-1	Región del este de EE. UU. (Ohio)	04:00 a 12:00 h UTC
us-west-1	Región del oeste de EE. UU (N. California)	06:00 — 17:00 UTC
us-west-2	Región del oeste de EE. UU (Oregon)	06:00 — 14:00 UTC

Cambio del periodo de mantenimiento del clúster

La ventana de mantenimiento debe corresponder al momento de mínimo uso y, por tanto, podría ser preciso modificarla cada cierto tiempo. Puede modificar el clúster de modo que especifique un

intervalo de tiempo de hasta 24 horas durante las cuales deban llevarse a cabo todas las actividades de mantenimiento que solicite. Las modificaciones de clúster pendientes o aplazadas que ha solicitado tendrán lugar en este periodo.

Más información

Para obtener más información sobre el periodo de mantenimiento y de la sustitución de nodos, consulte lo siguiente:

- Sustitución de nodos: administración de la sustitución de nodos
- Modificación de un clúster de MemoryDB: cambio del periodo de mantenimiento del clúster

Prácticas recomendadas

A continuación, puede encontrar las prácticas recomendadas para MemoryDB para Redis. Si las sigue, mejorará el rendimiento y la fiabilidad de su clúster.

Temas

- Comandos de Redis restringidos
- Resiliencia en MemoryDB para Redis
- Prácticas recomendadas: publicación/suscripción y multiplexación de E/S mejorada
- Prácticas recomendadas: redimensionamiento de clústeres en línea

Comandos de Redis restringidos

Para ofrecer una experiencia de servicio administrado, MemoryDB restringe el acceso a determinados comandos que requieren privilegios avanzados. Los siguientes comandos no están disponibles:

- `acl deluser`
- `acl load`
- `acl save`
- `acl setuser`
- `bgrewriteaof`
- `bgsave`
- `cluster addslot`
- `cluster delslot`
- `cluster setslot`
- `config`
- `debug`
- `migrate`
- `module`
- `psync`
- `replicaof`
- `save`
- `shutdown`
- `slaveof`
- `sync`

Resiliencia en MemoryDB para Redis

La infraestructura global de AWS se compone de regiones y zonas de disponibilidad de AWS. AWS Las regiones proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre zonas de disponibilidad sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

Para obtener más información sobre las regiones y zonas de disponibilidad de AWS, consulte Infraestructura global de AWS.

Además de la infraestructura global de AWS, MemoryDB para Redis ofrece varias características que le ayudan con sus necesidades de instantáneas y resiliencia de los datos.

Temas

Mitigación de errores

A la hora de planificar la implementación de MemoryDB para Redis, debe hacerlo de modo que los errores tengan una repercusión mínima en su aplicación y sus datos. Los temas de esta sección abordan enfoques que puede aplicar para proteger la aplicación y los datos frente a errores.

Mitigación de errores: clústeres de MemoryDB

Un clúster de MemoryDB se compone de un único nodo principal, disponible para operaciones de lectura y escritura para su aplicación y de 0 a 5 nodos de réplica de solo lectura. Sin embargo, recomendamos encarecidamente utilizar al menos una réplica para una alta disponibilidad. Cuando se escriben datos en el nodo principal, también se conservan en el registro de transacciones y de forma asíncrona en los nodos de réplica.

Qué sucede en caso de error en una réplica de lectura

1. MemoryDB detecta la réplica con error.
2. MemoryDB deja el nodo con error sin conexión.
3. MemoryDB ejecuta y aprovisiona un nodo de reemplazo en la misma zona de disponibilidad.

4. El nuevo nodo se sincroniza con el registro de transacciones.

Durante este tiempo, la aplicación podrá seguir realizando operaciones de lectura y escritura con los demás nodos.

MemoryDB Multi-AZ

Si el Multi-AZ está activado en sus clústeres de MemoryDB, se detectará un error en el primario y se reemplazará automáticamente.

1. MemoryDB detecta el error del nodo principal.
2. MemoryDB realiza una conmutación por error a una réplica después de asegurarse de que es coherente con la copia principal que ha fallado.
3. MemoryDB pone en marcha una réplica en la zona de disponibilidad del nodo principal con error.
4. El nuevo nodo se sincroniza con el nuevo nodo se sincroniza con el registro de transacciones.

La conmutación por error a un nodo de réplica suele ser más rápida que la creación y el aprovisionamiento de un nuevo nodo principal. Esto significa que la aplicación podrá reanudar la escritura en el nodo principal antes.

Para obtener más información, consulte Minimización del tiempo de inactividad en MemoryDB con Multi-AZ.

Prácticas recomendadas: publicación/suscripción y multiplexación de E/S mejorada

Cuando utilice la versión 7 o posterior de Redis, recomendamos utilizar publicación/envío fragmentado. También mejora el rendimiento y la latencia mediante la multiplexación de E/S mejorada, que está disponible automáticamente cuando se utiliza la versión 7 o posterior de Redis y no requiere cambios en el cliente. Es ideal para cargas de trabajo de publicación/envío, que suelen estar limitadas por rendimiento con múltiples conexiones de cliente.

Prácticas recomendadas: redimensionamiento de clústeres en línea

El cambio de las particiones implica agregar y eliminar particiones o nodos del clúster y redistribuir los espacios clave. Como resultado, varios aspectos influyen en la operación de cambio de las particiones, como la carga en el clúster, la utilización de memoria y el tamaño total de los datos. Para disfrutar de la mejor experiencia, recomendamos que siga las prácticas recomendadas de clúster global para una distribución uniforme del patrón de carga de trabajo. Además, recomendamos que siga los pasos que se detallan a continuación.

Antes de iniciar el cambio de las particiones, recomendamos lo siguiente:

- Probar la aplicación: si es posible, pruebe el comportamiento de la aplicación durante el cambio de las particiones en un entorno de ensayo.
- Recibir notificaciones anticipadas sobre problemas de escalado: el cambio de particiones es una operación que requiere mucho procesamiento. Por ello, recomendamos que mantenga el uso de la CPU por debajo del 80 por ciento en instancias de varios núcleos y en menos del 50 por ciento en instancias de un solo núcleo durante el cambio de particiones. Monitoree las métricas de MemoryDB e inicie el cambio de las particiones antes de que la aplicación comience a observar problemas de escalado. Las métricas de las que se puede realizar un seguimiento son `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections`, `NewConnections`, `FreeableMemory`, `SwapUsage` y `BytesUsedForMemoryDB`.
- Comprobar que hay suficiente memoria libre disponible antes de la reducción horizontal: si va a realizar una reducción horizontal, asegúrese de que la memoria libre disponible en las particiones que se van a retener sea al menos 1,5 veces la memoria utilizada en las particiones que tiene previsto eliminar.
- Iniciar el cambio de las particiones durante las horas de menor actividad: esta práctica contribuye a reducir la latencia y el impacto en el rendimiento en el cliente durante la operación de cambio de

las particiones. También ayuda a completar el cambio de las particiones con mayor rapidez ya que se pueden usar más recursos para la redistribución de ranuras.

- Revisar el comportamiento de tiempo de espera de cliente: es posible que algunos clientes observen una latencia más alta durante el cambio de tamaño del clúster en línea. La configuración de la biblioteca de cliente con un tiempo de espera más alto puede ayudar a conceder al sistema tiempo para conectar incluso en condiciones de carga más altas en servidor. En algunos casos, es posible que abra un gran número de conexiones al servidor. En estos casos, considere la posibilidad de agregar retardo exponencial a la lógica de reconexión. Si lo hace, puede ayudar a evitar que llegue una ráfaga de conexiones nuevas al servidor al mismo tiempo.

Durante el cambio de las particiones, recomendamos lo siguiente:

- Evitar los comandos costosos: evite ejecutar operaciones que hagan una utilización intensiva de procesamiento y de E/S, como los comandos `KEYS` y `SMEMBERS`. Recomendamos este enfoque porque estas operaciones aumentan la carga en el clúster e influyen en el rendimiento del clúster. En su lugar, utilice los comandos `SCAN` y `SSCAN`.
- Seguir las prácticas recomendadas de Lua: evite los scripts Lua de ejecución prolongada y siempre declare por adelantado las claves que utiliza en los scripts Lua. Recomendamos este enfoque para determinar que el script Lua no está utilizando comandos de ranura cruzada. Asegúrese de que las claves utilizadas en scripts Lua pertenezcan a la misma ranura.

Después del cambio de las particiones, tenga en cuenta lo siguiente:

- La reducción horizontal se puede realizar parcialmente si no hay suficiente memoria disponible en las particiones de destino. Si se produce este resultado, revise la memoria disponible y, si es necesario, reintente la operación.
- Las ranuras con elementos grandes no se migran. En concreto, no se migran las ranuras con elementos que superen los 256 MB después de la serialización.
- No se admiten los comandos `FLUSHALL` y `FLUSHDB` en los scripts Lua dentro durante una operación de cambio de particiones.

Descripción de cómo replicar en MemoryDB

MemoryDB implementa la reproducción con datos particionados en hasta 500 particiones.

Cada partición de un clúster tiene un nodo principal de lectura/escritura y hasta 5 nodos de réplica de solo lectura. Cada nodo principal puede soportar hasta 100 MB/s. Puede crear un clúster con un mayor número de particiones y un menor número de réplicas con un total de hasta 500 nodos por clúster. Esta configuración de clúster puede variar desde 500 particiones y 0 réplicas hasta 100 particiones y 4 réplicas, que es el número máximo de réplicas permitido.

Coherencia

En MemoryDB, los nodos principales son muy consistentes. Las operaciones de escritura correctas se almacenan de forma duradera en registros transaccionales distribuidos en zonas de disponibilidad múltiples (Multi-AZ) antes de devolverlas a los clientes. Las operaciones de lectura en los archivos primarios siempre devuelven los datos más actualizados, lo que refleja los efectos de todas las operaciones de escritura anteriores que se realizaron correctamente. Esta sólida coherencia se mantiene en todas las conmutaciones por error principales.

En MemoryDB, los nodos de réplica son eventualmente consistentes. Es posible que las operaciones de lectura de réplicas (mediante `READONLY` comandos) no siempre reflejen los efectos de las operaciones de escritura más recientes que se realizaron correctamente, ya que las métricas de retardo se publican en CloudWatch. Sin embargo, las operaciones de lectura de una única réplica son coherentes secuencialmente. Las operaciones de escritura correctas tienen efecto en cada réplica en el mismo orden en que se ejecutaron en la principal.

Replicación en un clúster

Cada réplica de lectura de una partición mantiene una copia de los datos del nodo principal de la partición. Se utilizan mecanismos de replicación asíncronos mediante registros de transacciones para mantener las réplicas de lectura sincronizadas con el principal. Las aplicaciones pueden leer de cualquier nodos del clúster. Las aplicaciones pueden escribir únicamente en los nodos. Las réplicas de lectura mejoran la escalabilidad de lectura. Como MemoryDB almacena los datos en registros de transacciones duraderos, no hay riesgo de que los datos se pierdan. Los datos están particionados en las distintas particiones del clúster de MemoryDB.

Las aplicaciones utilizan el punto de conexión del clúster de MemoryDB para conectarse a los nodos del clúster. Para obtener más información, consulte Búsqueda de puntos de conexión.

Los clústeres de MemoryDB son regionales y solo pueden contener nodos de una región. Para mejorar la tolerancia a errores, puede aprovisionar tanto los principales como las réplicas de lectura en varias zonas de disponibilidad dentro de esa región.

Se recomienda encarecidamente utilizar la replicación, que proporciona Multi-AZ, para todos los clústeres de MemoryDB. Para obtener más información, consulte Minimización del tiempo de inactividad en MemoryDB con Multi-AZ.

Minimización del tiempo de inactividad en MemoryDB con Multi-AZ

Hay varias situaciones en las que MemoryDB puede necesitar reemplazar un nodo principal. Entre ellas se incluyen determinados tipos de mantenimiento planificado y el caso poco probable de que se produzca un error en el nodo principal o en la zona de disponibilidad.

La respuesta al fallo del nodo depende del nodo que haya fallado. Sin embargo, en todos los casos, MemoryDB garantiza que no se pierdan datos durante la sustitución de nodos o la conmutación por error. Por ejemplo, si una réplica falla, el nodo fallido se reemplaza y los datos se sincronizan desde el registro de transacciones. Si el nodo principal falla, se desencadena una conmutación por error a una réplica coherente, lo que garantiza que no se pierdan datos durante la conmutación por error. Las escrituras ahora se realizan desde el nuevo nodo principal. A continuación, el nodo principal anterior se reemplaza y se sincroniza desde el registro de transacciones.

Si un nodo principal falla en una partición de un solo nodo (sin réplicas), MemoryDB deja de aceptar escrituras hasta que se sustituya el nodo principal y se sincronice desde el registro de transacciones.

El reemplazo de un nodo produce un tiempo de inactividad para el clúster, pero si Multi-AZ se encuentra activo, el tiempo de inactividad es mínimo. El rol del nodo principal tendrá una conmutación por error automática en una de las réplicas. No es necesario crear ni aprovisionar un nodo principal nuevo, ya que MemoryDB se encargará de esto de forma clara. Esta conmutación por error y promoción de réplica garantizan la posibilidad de reanudar la escritura en la réplica principal tan pronto como se complete la promoción.

En caso de que se inicien sustituciones de nodos planeadas debido a actualizaciones de mantenimiento o actualizaciones de servicio, tenga en cuenta que las sustituciones de nodos planeadas se completan mientras el clúster atiende las solicitudes de escritura entrantes.

Las zonas de disponibilidad múltiples en los clústeres de MemoryDB mejoran la tolerancia a los errores. Esto es cierto especialmente en los casos en que el nodo principal del clúster deja de estar accesible o de funcionar por cualquier motivo. La función Multi-AZ en los clústeres de MemoryDB requiere que cada partición tenga más de un nodo y se habilita automáticamente.

Temas

Escenarios de error con respuestas de Multi-AZ

Si Multi-AZ está activo, un nodo principal que produce error conmuta por error a una réplica disponible. La réplica se sincroniza automáticamente con el registro de transacciones y pasa a ser principal, lo que es mucho más rápido que crear y volver a aprovisionar un nodo principal nuevo. Este proceso suele tardar tan solo unos segundos hasta que se puede escribir de nuevo en el clúster.

Cuando Multi-AZ está activo, MemoryDB monitorea continuamente el estado del nodo principal. Si se produce un error en el nodo principal, se realiza una de las siguientes acciones en función del tipo de error.

Temas

Escenarios de error cuando solo se produce un error en el nodo principal

Si solo se produce un error en el nodo principal, la réplica se convertirá automáticamente en principal. A continuación, se crea una réplica de reemplazo y se aprovisiona en la misma zona de disponibilidad que el principal ha producido un error.

Cuando solo se produce un error en el nodo principal, Multi-AZ de MemoryDB hace lo siguiente:

1. El nodo principal con error se desconecta (sin conexión).
2. Una réplica actualizada se convierte automáticamente en principal.

   Las operaciones de escritura se pueden reanudar tan pronto como se haya completado el proceso de conmutación por error, por lo general, en tan solo unos segundos.
3. Una réplica de reemplazo se lanza y aprovisiona.

   La réplica de reemplazo se lanza en la zona de disponibilidad en la que estaba el nodo principal con error, por lo que se mantiene la distribución de los nodos.
4. La réplica se sincroniza con el registro de transacciones.

Para obtener información acerca de la búsqueda de los puntos de conexión de un clúster, consulte los temas siguientes:

- Búsqueda del punto de conexión para un clúster de MemoryDB (API de MemoryDB)

Escenarios de error cuando el nodo principal y algunas réplicas producen un error

Si se produce un error en el nodo principal y en al menos una réplica, una réplica actualizada se promocionará al clúster principal. Las nuevas réplicas también se crean y se aprovisionan en las mismas zonas de disponibilidad que las de los nodos con error.

Cuando el nodo principal y algunas réplicas producen un error, Multi-AZ de MemoryDB hace lo siguiente:

1. El nodo principal y las réplicas con error se desconectan.
2. Una réplica disponible se convertirá en el nodo principal.

   Las operaciones de escritura se pueden reanudar en cuanto se haya completado el proceso de conmutación por error, por lo general, en tan solo unos segundos.
3. Las réplicas de reemplazo se crean y se aprovisionan.

   Las réplicas de reemplazo se crean en las zonas de disponibilidad de los nodos con error para, de este modo, conservar la distribución de los nodos.
4. Todos los nodos se sincronizan con el registro de transacciones.

Para obtener información acerca de la búsqueda de los puntos de conexión de un clúster, consulte los temas siguientes:

- Búsqueda del punto de conexión para un clúster de MemoryDB (CLI de AWS)
- Búsqueda del punto de conexión para un clúster de MemoryDB (API de MemoryDB)

Escenarios de error cuando se produce un error en todo el clúster

Si el error es general, todos los nodos se volverán a crear y a aprovisionar en las mismas zonas de disponibilidad que las de los nodos originales.

No hay pérdida de datos en este escenario, ya que los datos se conservaban en el registro de transacciones.

Cuando se produce un error en todo el clúster, Multi-AZ de MemoryDB hace lo siguiente:

1. El nodo principal y las réplicas se desconectan.
2. Se crea y aprovisiona un nodo principal de reemplazo, que se sincroniza con el registro de transacciones.
3. Se crean y aprovisionan réplicas de reemplazo, sincronizándolas con el registro de transacciones.

   Los reemplazos se crean en las zonas de disponibilidad de los nodos con error para, de este modo, conservar la distribución de los nodos.

Para obtener información acerca de la búsqueda de los puntos de conexión de un clúster, consulte los temas siguientes:

- Búsqueda del punto de conexión para un clúster de MemoryDB (CLI de AWS)
- Búsqueda del punto de conexión para un clúster de MemoryDB (API de MemoryDB)

Prueba de la conmutación por error automática

Puede probar la conmutación por error automática mediante la consola de MemoryDB, la AWS CLI y la API de MemoryDB.

Cuando realice las pruebas, tenga en cuenta lo siguiente:

- Puede utilizar esta operación hasta cinco veces en un periodo de 24 horas.
- Si realiza una llamada a esta operación en particiones de distintos clústeres, puede realizar las llamadas de forma simultánea.
- En algunos casos, es posible llamar a esta operación varias veces en particiones diferentes del mismo clúster de MemoryDB. En tales casos, la sustitución del primer nodo debe completarse antes de que se pueda realizar una llamada posterior.
- Para determinar si la sustitución del nodo se ha completado, consulte los eventos mediante la consola de MemoryDB para Redis, la AWS CLI o la API de MemoryDB. Busque los siguientes eventos relacionados con `FailoverShard`, que se indican a continuación por orden de incidencia:

  1. mensaje de clúster: `FailoverShard API called for shard <shard-id>`
  2. mensaje de clúster: `Failover from primary node <primary-node-id> to replica node <node-id> completed`
  3. mensaje de clúster: `Recovering nodes <node-id>`
  4. mensaje de clúster: `Finished recovery for nodes <node-id>`

  Para obtener más información, consulte lo siguiente:

  - DescribeEvents en la Referencia de la API de MemoryDB
- Esta API se ha diseñado para probar el comportamiento de la aplicación en caso de conmutación por error de MemoryDB. No está diseñado para ser una herramienta operativa para iniciar una conmutación por error para solucionar un problema con el clúster. Además, en determinadas condiciones, como los acontecimientos operacionales a gran escala, AWS puede bloquear esta API.

Temas

- Prueba de la conmutación por error automática mediante la AWS Management Console
- Prueba de la conmutación por error automática mediante la AWS CLI
- Prueba de la conmutación por error automática mediante la API de MemoryDB

Prueba de la conmutación por error automática mediante la AWS Management Console

Utilice el procedimiento siguiente para probar la conmutación por error automática con la consola.

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. Seleccione el botón de opción situado a la izquierda al clúster que desea probar. Este clúster debe tener al menos un nodo de réplica.
3. En el área Details, asegúrese de que este clúster tiene habilitadas Multi-AZ. Si el clúster no tiene habilitado Multi-AZ, elija un clúster distinto o modifique este clúster para habilitar Multi-AZ. Para obtener más información, consulte Modificación de un clúster de MemoryDB.
4. Elija el nombre del clúster.
5. En la página Particiones y nodos, elija el nombre de la partición en la que desea probar la conmutación por error.
6. Para el nodo, elija Realizar conmutación por error del nodo principal.
7. Elija Continue para realizar la conmutación por error al nodo principal, o bien Cancel para cancelar la operación y no realizar la conmutación por error al nodo principal.

   Durante el proceso de conmutación por error, la consola seguirá mostrando el estado del nodo como disponible. Para realizar un seguimiento del progreso de la prueba de la conmutación por error, elija Events en el panel de navegación de la consola. En la pestaña Eventos, consulte los eventos que indican que la conmutación por error se ha iniciado (`FailoverShard API called`) y completado (`Recovery completed`).

Prueba de la conmutación por error automática mediante la AWS CLI

Puede probar la conmutación por error automática en cualquier clúster habilitado de Multi-AZ mediante la operación de la AWS CLI failover-shard.

Parámetros

- `--cluster-name`: obligatorio. El clúster que se va a probar.
- `--shard-name`: obligatorio. Nombre de la partición en la que desea probar la conmutación por error automática. Puede probar un máximo de cinco particiones en un periodo de 24 horas.

En el siguiente ejemplo, se utiliza la AWS CLI para realizar una llamada a `failover-shard` en la partición `0001` del clúster de MemoryDB `my-cluster`.

Para Linux, macOS o Unix:

```
aws memorydb failover-shard \
   --cluster-name my-cluster \
   --shard-name 0001
```

Para Windows:

```
aws memorydb failover-shard ^
   --cluster-name my-cluster ^
   --shard-name 0001
```

Para realizar un seguimiento del progreso de la conmutación por error, use la operación de la AWS CLI `describe-events`.

Devuelve la siguiente respuesta JSON:

```
{
    "Events": [
        {
            "SourceName": "my-cluster",
            "SourceType": "cluster",
            "Message": "Failover to replica node my-cluster-0001-002 completed",
            "Date": "2021-08-22T12:39:37.568000-07:00"
        },
        {
            "SourceName": "my-cluster",
            "SourceType": "cluster",
            "Message": "Starting failover for shard 0001",
            "Date": "2021-08-22T12:39:10.173000-07:00"
        }
    ]
}
```

Para obtener más información, consulte lo siguiente:

- failover-shard

- describe-events

Prueba de la conmutación por error automática mediante la API de MemoryDB

En el siguiente ejemplo, se realiza una llamada a `FailoverShard` en la partición `0003` del clúster `memorydb00`.

Example Prueba de la conmutación por error automática

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=FailoverShard
    &ShardName=0003
    &ClusterName=memorydb00
    &Version=2021-01-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210801T192317Z
    &X-Amz-Credential=<credential>
```

Para realizar un seguimiento del progreso de la conmutación por error, use la operación `DescribeEvents` de la API de MemoryDB.

Para obtener más información, consulte lo siguiente:

- FailoverShard
- DescribeEvents

Cambio del número de réplicas

Puede aumentar o disminuir dinámicamente el número de réplicas de lectura del clúster de MemoryDB mediante la AWS Management Console, la AWS CLI o la API de MemoryDB. Todas las particiones deben tener el mismo número de réplicas.

Aumento del número de réplicas de un clúster

Puede aumentar el número de réplicas de un clúster de MemoryDB hasta un máximo de cinco por partición. Para ello, utilice la AWS Management Console, la AWS CLI o la API de MemoryDB.

Temas

- Utilización de la AWS Management Console
- Utilización de la AWS CLI
- Uso de la API de MemoryDB

Utilización de la AWS Management Console

Para aumentar el número de réplicas en un clúster de MemoryDB (consola), consulte Agregar/eliminar nodos de un clúster.

Utilización de la AWS CLI

Para aumentar el número de réplicas de un clúster de MemoryDB, utilice el comando `update-cluster` con los parámetros siguientes:

- `--cluster-name`: obligatorio. Identifica el clúster en el que desea aumentar el número de réplicas.
- `--replica-configuration`: obligatorio. Le permite establecer el número de réplicas. Para aumentar el número de réplicas, establezca la `ReplicaCount` propiedad en el número de réplicas que desea incluir en la partición al final de la operación.

Example

En el siguiente ejemplo, se aumenta el número de réplicas del clúster `my-cluster` a 2.

Para Linux, macOS o Unix:

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --replica-configuration \
        ReplicaCount=2
```

Para Windows:

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
    --replica-configuration ^
        ReplicaCount=2
```

Devuelve la siguiente respuesta JSON:

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "updating",
        "NumberOfShards": 1,
        "ClusterEndpoint": {
            "Address": "clustercfg.my-cluster.xxxxx.memorydb.us-east-1.amazonaws.com",
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "DataTiering": "false",
        "AutoMinorVersionUpgrade": true
    }
}
```

Para ver los detalles del clúster actualizado una vez que su estado cambie de actualizado a disponible, utilice el siguiente comando:

Para Linux, macOS o Unix:

```
aws memorydb describe-clusters \
    --cluster-name my-cluster
    --show-shard-details
```

Para Windows:

```
aws memorydb describe-clusters ^
    --cluster-name my-cluster
    --show-shard-details
```

Devuelve la siguiente respuesta JSON:

```
{
    "Clusters": [
        {
            "Name": "my-cluster",
            "Status": "available",
            "NumberOfShards": 1,
            "Shards": [
                {
                    "Name": "0001",
                    "Status": "available",
                    "Slots": "0-16383",
                    "Nodes": [
                        {
                            "Name": "my-cluster-0001-001",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        },
                        {
                            "Name": "my-cluster-0001-002",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1b",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        },
                        {
                            "Name": "my-cluster-0001-003",
```

```
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": "2021-08-22T12:59:31.844000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        }
                    ],
                    "NumberOfNodes": 3
                }
            ],
            "ClusterEndpoint": {
                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                "Port": 6379
            },
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "ParameterGroupName": "default.memorydb-redis6",
            "ParameterGroupStatus": "in-sync",
            "SubnetGroupName": "my-sg",
            "TLSEnabled": true,
            "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
            "SnapshotRetentionLimit": 0,
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "SnapshotWindow": "04:30-05:30",
            "ACLName": "my-acl",
            "DataTiering": "false",
            "AutoMinorVersionUpgrade": true
        }
    ]
}
```


Para obtener más información acerca de cómo aumentar el número de réplicas mediante la CLI, consulte update-cluster en la Referencia de comandos de AWS CLI.

Uso de la API de MemoryDB

Para aumentar el número de réplicas de una partición de MemoryDB, utilice la acción `UpdateCluster` con los parámetros siguientes:

- `ClusterName`: obligatorio. Identifica el clúster en el que desea aumentar el número de réplicas.
- `ReplicaConfiguration`: obligatorio. Le permite establecer el número de réplicas. Para aumentar el número de réplicas, establezca la `ReplicaCount` propiedad en el número de réplicas que desea incluir en la partición al final de la operación.

Example

En el siguiente ejemplo, se aumenta el número de réplicas del clúster `sample-cluster` a tres. Al finalizar el ejemplo, existirán tres réplicas en cada partición. Este número se aplica tanto si se trata de un clúster de MemoryDB con una única partición como de un clúster de MemoryDB con varias particiones.

```
https://memory-db.us-east-1.amazonaws.com/
      ?Action=UpdateCluster
      &ReplicaConfiguration.ReplicaCount=3
      &ClusterName=sample-cluster
      &Version=2021-01-01
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20210802T192317Z
      &X-Amz-Credential=<credential>
```

Para obtener más información acerca de cómo aumentar el número de réplicas mediante la API, consulte UpdateCluster.

Reducción del número de réplicas de un clúster

Puede reducir el número de réplicas de una partición de MemoryDB. Puede reducir el número de réplicas a cero, pero no puede realizar una conmutación por error a una réplica si el nodo principal falla.

Puede utilizar la AWS Management Console, la AWS CLI o la API de MemoryDB para reducir el número de réplicas de un clúster.

Temas

Utilización de la AWS Management Console

Para reducir el número de réplicas en un clúster de MemoryDB (consola), consulte Agregar/eliminar nodos de un clúster.

Utilización de la AWS CLI

Para reducir el número de réplicas de un clúster de MemoryDB, utilice el comando `update-cluster` con los parámetros siguientes:

- `--cluster-name`: obligatorio. Identifica el clúster en el que se desea reducir el número de réplicas.
- `--replica-configuration`: obligatorio.

  `ReplicaCount`: defina esta propiedad para especificar el número de nodos de réplica que desea.

Example

En el siguiente ejemplo, se utiliza `--replica-configuration` para reducir el número de réplicas del clúster `my-cluster` al valor especificado.

Para Linux, macOS o Unix:

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --replica-configuration \
```

```
        ReplicaCount=1
```

Para Windows:

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
    --replica-configuration ^
        ReplicaCount=1 ^
```

Devuelve la siguiente respuesta JSON:

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "updating",
        "NumberOfShards": 1,
        "ClusterEndpoint": {
            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-east-1.amazonaws.com",
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "DataTiering": "false",
        "AutoMinorVersionUpgrade": true
    }
}
```

Para ver los detalles del clúster actualizado una vez que su estado cambie de actualizado a disponible, utilice el siguiente comando:

Para Linux, macOS o Unix:

```
aws memorydb describe-clusters \
```

```
    --cluster-name my-cluster
    --show-shard-details
```

Para Windows:

```
aws memorydb describe-clusters ^
    --cluster-name my-cluster
    --show-shard-details
```

Devuelve la siguiente respuesta JSON:

```
{
    "Clusters": [
        {
            "Name": "my-cluster",
            "Status": "available",
            "NumberOfShards": 1,
            "Shards": [
                {
                    "Name": "0001",
                    "Status": "available",
                    "Slots": "0-16383",
                    "Nodes": [
                        {
                            "Name": "my-cluster-0001-001",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        },
                        {
                            "Name": "my-cluster-0001-002",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1b",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
```

```
                                "Port": 6379
                            }
                        }
                    ],
                    "NumberOfNodes": 2
                }
            ],
            "ClusterEndpoint": {
                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                "Port": 6379
            },
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "ParameterGroupName": "default.memorydb-redis6",
            "ParameterGroupStatus": "in-sync",
            "SubnetGroupName": "my-sg",
            "TLSEnabled": true,
            "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
            "SnapshotRetentionLimit": 0,
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "SnapshotWindow": "04:30-05:30",
            "ACLName": "my-acl",
            "DataTiering": "false",
            "AutoMinorVersionUpgrade": true
        }
    ]
}
```

Para obtener más información acerca de cómo reducir el número de réplicas mediante la CLI, consulte update-cluster en la Referencia de comandos de AWS CLI.

Uso de la API de MemoryDB

Para reducir el número de réplicas de un clúster de MemoryDB, utilice la acción `UpdateCluster` con los parámetros siguientes:

- `ClusterName`: obligatorio. Identifica el clúster en el que se desea reducir el número de réplicas.
- `ReplicaConfiguration`: obligatorio. Le permite establecer el número de réplicas.

  `ReplicaCount`: defina esta propiedad para especificar el número de nodos de réplica que desea.

Example

En el siguiente ejemplo, se utiliza `ReplicaCount` para reducir el número de réplicas del clúster `sample-cluster` a una. Al finalizar el ejemplo, existirá una réplica en cada partición. Este número se aplica tanto si se trata de un clúster de MemoryDB con una única partición como de un clúster de MemoryDB con varias particiones.

```
https://memory-db.us-east-1.amazonaws.com/
      ?Action=UpdateCluster
      &ReplicaConfiguration.ReplicaCount=1
      &ClusterName=sample-cluster
      &Version=2021-01-01
      &SignatureVersion=4
      &SignatureMethod=HmacSHA256
      &Timestamp=20210802T192317Z
      &X-Amz-Credential=<credential>
```

Para obtener más información acerca de cómo reducir el número de réplicas mediante la API, consulte UpdateCluster.

Instantánea y restauración

Los clústeres de MemoryDB para Redis realizan copias de seguridad automáticas de los datos en un registro transaccional Multi-AZ, pero puede optar por crear instantáneas puntuales de un clúster de forma periódica o bajo demanda. Estas instantáneas se pueden usar para recrear un clúster en un momento anterior o para generar un clúster completamente nuevo. La instantánea se compone de los metadatos del clúster, junto con todos los datos del clúster. Todas las instantáneas se escriben en Amazon Simple Storage Service (Amazon S3), lo que proporciona un almacenamiento duradero. En cualquier momento, puede restaurar los datos creando un nuevo clúster de MemoryDB y rellenándolo con los datos de una instantánea. Con MemoryDB, puede administrar las instantáneas mediante las APIAWS Management Console, AWS Command Line Interface (AWS CLI) y MemoryDB.

Temas

- Restricciones relativas a las instantáneas
- Costos de las instantáneas
- Programación de instantáneas automáticas
- Toma de instantáneas manuales

- Creación de una instantánea final
- Descripción de instantáneas
- Copia de una instantánea
- Exportación de instantáneas
- Restauración a partir de una instantánea
- Inicialización de un nuevo clúster con una instantánea creada externamente
- Etiquetado de instantáneas
- Eliminación de una instantánea

Restricciones relativas a las instantáneas

Debe tener en cuenta las limitaciones siguientes a la hora de planear o realizar instantáneas:

- En el caso de los clústeres de MemoryDB, las instantáneas y la restauración están disponibles para todos los tipos de nodos compatibles.
- Durante un periodo de 24 horas continuas, no podrá crear más de 20 instantáneas manuales por clúster.
- MemoryDB solo admite la toma de instantáneas a nivel de clúster. MemoryDB no admite la toma de instantáneas a nivel de partición o nodo.
- Durante el proceso de instantánea, no podrá realizar operaciones de la API o la CLI en el clúster.
- Si elimina un clúster y solicita una instantánea final, MemoryDB siempre realizará la instantánea de los nodos principales. De este modo, se garantiza que se capturan los datos más recientes antes de eliminar el clúster.

Costos de las instantáneas

MemoryDB permite almacenar una instantánea por cada clúster de MemoryDB activo de forma gratuita. El espacio de almacenamiento para instantáneas adicionales se cobra a una tarifa de 0,085 USD por GB al mes para todas las regiones de AWS. No se aplican tarifas de transferencia de datos para la creación de instantáneas o para la restauración de datos de una instantánea a un clúster de MemoryDB.

Programación de instantáneas automáticas

Para cualquier clúster de MemoryDB, puede habilitar las instantáneas automáticas. Cuando se habilitan las instantáneas automáticas, MemoryDB crea una instantánea del clúster una vez al día. No hay impacto en el clúster y el cambio es inmediato. Para obtener más información, consulte Restauración a partir de una instantánea.

Al programar instantáneas automáticas, debe planificar los ajustes siguientes:

- Periodo de instantáneas: periodo del día durante el cual MemoryDB comienza a crear una instantánea. La duración mínima para el periodo de instantáneas es de 60 minutos. Puede configurar el periodo de instantáneas a la hora que más le convenga o a una hora del día a la que la instantánea no se realice en periodos de uso especialmente intensivos.

  Si no especifica ningún periodo de instantáneas, MemoryDB asignará uno automáticamente.
- Límite de retención de instantánea: número de días que se retiene la instantánea en Amazon S3. Por ejemplo, si establece el límite de retención en 5, una instantánea que se realice hoy se conservaría durante 5 días. Al finalizar el límite de retención, la instantánea se eliminará automáticamente.

  El límite máximo de retención de instantánea es de 35 días. Si el límite de retención de instantánea se establece en 0, las instantáneas se deshabilitarán en el clúster. Los datos de MemoryDB siguen siendo totalmente duraderos incluso con la captura automática de instantáneas desactivada.

Puede habilitar o deshabilitar las instantáneas automáticas durante la creación de un clúster de MemoryDB mediante la consola de MemoryDB, la AWS CLI o la API de MemoryDB. Puede activar las instantáneas automáticas al crear un clúster de MemoryDB marcando la casilla Habilitar copias de seguridad automáticas en la sección Instantáneas. Para obtener más información, Creación de un clúster de MemoryDB.

Toma de instantáneas manuales

Además de las instantáneas automáticas, puede crear una instantánea manual en cualquier momento. A diferencia de las instantáneas automáticas, que se eliminan automáticamente después de un periodo de retención determinado, las instantáneas manuales no tienen periodo de retención que determine su eliminación automática. Las instantáneas manuales deben eliminarse manualmente. Incluso si elimina un clúster o un nodo, las instantáneas manuales de dicho clúster o nodo se conservarán. Si ya no desea conservar una instantánea manual, deberá eliminarla de forma explícita.

Las instantáneas manuales son útiles para el archivado y la realización de pruebas. Por ejemplo, supongamos que ha desarrollado un conjunto de datos de base para realizar distintas pruebas. Puede crear una instantánea manual de los datos y restaurarla siempre que lo desee. Tras probar la aplicación que modifica los datos, podrá restablecer los datos creando un nuevo clúster y restaurando los datos desde la instantánea de base. Cuando el clúster esté listo, podrá probar sus aplicaciones de nuevo con los datos de base y repetir este proceso tantas veces como sea necesario.

Además de crear directamente una instantánea manual, puede crear instantáneas manuales de las maneras siguientes:

- Copia de una instantánea: no importa si la instantánea de origen se creó automáticamente o manualmente.
- Creación de una instantánea final: cree una instantánea inmediatamente antes de eliminar un clúster.

Otros temas de importancia

- Restricciones relativas a las instantáneas
- Costos de las instantáneas

Puede crear una instantánea manual de un nodo mediante la AWS Management Console, la AWS CLI o la API de MemoryDB.

Creación de una instantánea manual (consola)

Para crear una instantánea de un clúster (consola)

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación izquierdo, elija Clústeres.

   Aparece la pantalla de clústeres de MemoryDB.
3. elija el botón de opción situado a la izquierda del nombre del clúster de MemoryDB del que desea realizar una copia de seguridad.
4. Elija Acciones y, a continuación, Tomar instantánea.
5. En la ventana Instantánea, escriba un nombre para la instantánea en el cuadro Nombre de la instantánea. Recomendamos que el nombre indique el clúster del que se hizo una copia de seguridad y especifique la fecha y la hora en que se creó la instantánea.

   Las restricciones para la asignación de nombres de clúster son las siguientes:

   - Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
   - Deben comenzar por una letra.
   - No pueden contener dos guiones consecutivos.
   - No pueden terminar con un guion.
6. En Cifrado, elija si desea usar la clave de cifrado predeterminada o una clave administrada por el cliente. Para obtener más información, consulte Cifrado en tránsito (TLS) de MemoryDB.
7. En Etiquetas, puede agregar etiquetas, de forma opcional, para buscar y filtrar las instantáneas o hacer un seguimiento de AWS los costos.
8. Elija Take Snapshot (Realizar una instantánea).

   El estado del clúster cambia a snapshotting. Cuando el estado vuelva a ser disponible, la instantánea se habrá realizado.

Creación de una instantánea manual (CLI de AWS)

Para crear una instantánea manual de un clúster mediante la AWS CLI, use la operación de la AWS CLI `create-snapshot` con los parámetros siguientes:

- `--cluster-name`: nombre del clúster de MemoryDB que se utilizará como fuente de la instantánea. Utilice este parámetro para realizar copias de seguridad de un clúster de MemoryDB.

  Las restricciones para la asignación de nombres de clúster son las siguientes:

  - Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
  - Deben comenzar por una letra.
  - No pueden contener dos guiones consecutivos.
  - No pueden terminar con un guion.

- `--snapshot-name`: nombre de la instantánea que se creará.

Temas relacionados de

Para obtener más información, consulte `create-snapshot` en la Referencia de los comandos de AWS CLI.

Creación de una instantánea manual (API de MemoryDB)

Para crear una instantánea manual de un clúster mediante la API de MemoryDB, use la operación de la API de MemoryDB `CreateSnapshot` con los parámetros siguientes:

- `ClusterName`: nombre del clúster de MemoryDB que se utilizará como fuente de la instantánea. Utilice este parámetro para realizar copias de seguridad de un clúster de MemoryDB.

  Las restricciones para la asignación de nombres de clúster son las siguientes:

  - Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
  - Deben comenzar por una letra.
  - No pueden contener dos guiones consecutivos.
  - No pueden terminar con un guion.
- `SnapshotName`: nombre de la instantánea que se creará.

Temas relacionados de

Para obtener más información, consulte CreateSnapshot.

Creación de una instantánea final

Puede crear una instantánea final con la consola de MemoryDB, la AWS CLI o la API de MemoryDB

Creación de una instantánea final (consola)

Puede crear una instantánea final al eliminar un clúster de MemoryDB mediante la consola de MemoryDB.

Para crear una instantánea final al eliminar un clúster de MemoryDB, en la página de eliminación, seleccione Sí y asigne un nombre a la instantánea en Paso 4: eliminar un clúster.

Creación de una instantánea final (AWSCLI)

Puede crear una instantánea final al eliminar un clúster de MemoryDB mediante la AWS CLI.

Al eliminar un clúster de MemoryDB

Para crear una instantánea final al eliminar un clúster, use la operación de la AWS CLI `delete-cluster` con los parámetros siguientes:

- `--cluster-name`: nombre del clúster que va a eliminar.
- `--final-snapshot-name`: nombre de la instantánea final.

El siguiente código toma la instantánea final `bkup-20210515-final` al eliminar el clúster `myCluster`.

Para Linux, macOS o Unix:

```
aws memorydb delete-cluster \
        --cluster-name myCluster \
        --final-snapshot-name bkup-20210515-final
```

Para Windows:

```
aws memorydb delete-cluster ^
        --cluster-name myCluster ^
        --final-snapshot-name bkup-20210515-final
```

Para obtener más información, consulte delete-cluster en la Referencia de comandos de la AWS CLI.

Creación de una instantánea final (API de MemoryDB)

Puede crear una instantánea final al eliminar un clúster de MemoryDB mediante la API de MemoryDB.

Al eliminar un clúster de MemoryDB

Para crear una instantánea final, use la operación de la API de MemoryDB `DeleteCluster` con los parámetros siguientes.

- `ClusterName`: nombre del clúster que va a eliminar.
- `FinalSnapshotName`: nombre de la instantánea.

La siguiente operación de la API de MemoryDB crea la instantánea `bkup-20210515-final` al eliminar el clúster `myCluster`.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DeleteCluster
    &ClusterName=myCluster
    &FinalSnapshotName=bkup-20210515-final
    &Version=2021-01-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210515T192317Z
    &X-Amz-Credential=<credential>
```

Para obtener más información, consulte DeleteCluster.

Descripción de instantáneas

Los siguientes procedimientos muestran cómo mostrar una lista de sus instantáneas. Si lo desea, también puede ver los detalles de una instantánea determinada.

Descripción de instantáneas (consola)

Para mostrar las instantáneas mediante el AWS Management Console

1. Inicie sesión en la consola
2. en el panel de navegación izquierdo, elija Instantáneas.
3. Utilice la búsqueda para filtrar las instantáneas manuales, automáticas o todas.
4. Para ver los detalles de una instantánea en particular, elija el botón de opción situado a la izquierda del nombre de la instantánea. Elija Acciones y, a continuación, Ver detalles.
5. Si lo desea, en la página Ver detalles, puede realizar acciones adicionales de la instantánea, como copiar, restaurar o eliminar. También puede agregar etiquetas a la instantánea.

Descripción de las instantáneas (AWSCLI)

Para mostrar una lista de las instantáneas y, de forma opcional, los detalles específicos de una instantánea, use la operación de la CLI `describe-snapshots`.

Ejemplos

La siguiente operación usa el parámetro `--max-results` para mostrar hasta 20 instantáneas asociadas a su cuenta. Si se omite el parámetro `--max-results` se muestran hasta 50 instantáneas.

```
aws memorydb describe-snapshots --max-results 20
```

La operación siguiente usa el parámetro `--cluster-name` para mostrar solo las instantáneas asociadas al clúster `my-cluster`.

```
aws memorydb describe-snapshots --cluster-name my-cluster
```

La siguiente operación usa el parámetro `--snapshot-name` para mostrar los detalles de la instantánea `my-snapshot`.

```
aws memorydb describe-snapshots --snapshot-name my-snapshot
```

Para obtener más información, consulte describe-snapshots.

Descripción de las instantáneas (API de MemoryDB)

Para mostrar una lista de las instantáneas, use la operación `DescribeSnapshots`.

Ejemplos

La siguiente operación usa el parámetro `MaxResults` para mostrar hasta 20 instantáneas asociadas a su cuenta. Si se omite el parámetro `MaxResults` se muestran hasta 50 instantáneas.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DescribeSnapshots
    &MaxResults=20
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Timestamp=20210801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210801T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

La operación siguiente usa el parámetro `ClusterName` para mostrar todas las instantáneas asociadas al clúster `MyCluster`.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DescribeSnapshots
    &ClusterName=MyCluster
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Timestamp=20210801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210801T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Credential=<credential>
```

```
    &X-Amz-Signature=<signature>
```

La siguiente operación usa el parámetro `SnapshotName` para mostrar los detalles de la instantánea `MyBackup`.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DescribeSnapshots
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &SnapshotName=MyBackup
    &Timestamp=20210801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210801T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información, consulte DescribeSnapshots.

Copia de una instantánea

Puede realizar una copia de cualquier instantánea, independientemente de si se creó de forma automática o manual. Al copiar una instantánea, se utiliza para el destino la misma clave de cifrado KMS que la fuente, a menos que se anule específicamente. También puede exportar una instantánea para poder obtener acceso a ella desde fuera de MemoryDB. Para obtener instrucciones acerca de cómo exportar su instantánea, consulte Exportación de instantáneas.

Los siguientes procedimientos muestran cómo copiar una instantánea.

Copia de una instantánea (consola)

Para copiar una instantánea (consola)

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. Para ver una lista de las instantáneas, en el panel de navegación izquierdo, elija Instantáneas.
3. En la lista de instantáneas, elija el botón de opción situado a la izquierda de la instantánea que desea copiar.
4. Elija Acciones y, a continuación, elija Copiar.
5. En la página Copiar instantánea, haga lo siguiente:

    a. En el cuadro Nuevo nombre de instantánea, especifique un nombre para la nueva instantánea.

    b. Deje el cuadro opcional Target S3 Bucket en blanco. Este campo solo debe usarse para exportar su instantánea y requiere permisos de S3 especiales. Para obtener información acerca de la exportación de instantáneas, consulte Exportación de instantáneas.

    c. Elija si desea utilizar la clave de AWS KMS cifrado predeterminada o utilizar una clave personalizada. Para obtener más información, consulte Cifrado en tránsito (TLS) de MemoryDB.

    d. De forma opcional, también puede agregar etiquetas a la copia instantánea.

    e. Elija Copiar.

Copia de una instantánea (CLI de AWS)

Para copiar una instantánea, use la operación `copy-snapshot`.

Parameters

- `--source-snapshot-name`: nombre de la instantánea que se copiará.
- `--target-snapshot-name`: nombre de la copia de la instantánea.
- `--target-bucket`: reservado para la exportación de una instantánea. No use este parámetro al realizar una copia de una instantánea. Para obtener más información, consulte Exportación de instantáneas.

El ejemplo siguiente realiza una copia de una instantánea automática.

Para Linux, macOS o Unix:

```
aws memorydb copy-snapshot \
    --source-snapshot-name automatic.my-primary-2021-03-27-03-15 \
    --target-snapshot-name my-snapshot-copy
```

Para Windows:

```
aws memorydb copy-snapshot ^
    --source-snapshot-name automatic.my-primary-2021-03-27-03-15 ^
    --target-snapshot-name my-snapshot-copy
```

Para obtener más información, consulte copy-snapshot.

Copiar una instantánea (API de MemoryDB)

Para copiar una instantánea, use la operación `copy-snapshot` con los parámetros siguientes:

Parameters

- `SourceSnapshotName`: nombre de la instantánea que se copiará.
- `TargetSnapshotName`: nombre de la copia de la instantánea.
- `TargetBucket`: reservado para la exportación de una instantánea. No use este parámetro al realizar una copia de una instantánea. Para obtener más información, consulte Exportación de instantáneas.

El ejemplo siguiente realiza una copia de una instantánea automática.

Example

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=CopySnapshot
    &SourceSnapshotName=automatic.my-primary-2021-03-27-03-15
    &TargetSnapshotName=my-snapshot-copy
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210801T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información, consulte CopySnapshot.

Exportación de instantáneas

MemoryDB para Redis permite exportar su instantánea de MemoryDB a un bucket de Amazon Simple Storage Service (Amazon S3), lo que facilita el acceso a la instantánea desde fuera de MemoryDB. Las instantáneas de MemoryDB exportadas son totalmente compatibles con Redis de código abierto y se pueden cargar con la versión o las herramientas de Redis adecuadas. Puede exportar una instantánea mediante la consola de MemoryDB, la AWS CLI o la API de MemoryDB

Exportar una instantánea puede resultar útil si necesita lanzar un clúster en otra región de AWS. De este modo, puede exportar sus datos en una región de AWS, copiar el archivo .rdb a la nueva región de AWS y, a continuación, utilizar dicho archivo .rdb para inicializar los datos en el nuevo clúster, en lugar de esperar a que se rellene con el uso. Para obtener información acerca de la propagación de datos en un nuevo clúster, consulte Inicialización de un nuevo clúster con una instantánea creada externamente. Otro motivo por el que es posible que desee exportar los datos de su clúster es para usar el archivo .rdb para el procesamiento sin conexión.

Important

- La instantánea de MemoryDB y el bucket de Amazon S3 en el que desea copiarla deben encontrarse en la misma región de AWS.

  Aunque las instantáneas copiadas en un bucket de Amazon S3 se encuentran cifradas, recomendamos encarecidamente que no conceda a otras personas acceso al bucket de Amazon S3 en el que desea almacenar las instantáneas.
- La exportación de una instantánea a Amazon S3 no se admite en clústeres que utilizan la organización de datos en niveles. Para obtener más información, consulte Organización de datos en niveles.

Para poder exportar una instantánea a un bucket de Amazon S3, debe disponer de un bucket de Amazon S3 en la misma región de AWS que la instantánea. Conceda a MemoryDB acceso al bucket. Los primeros dos pasos muestran cómo realizar esto último.

Warning

Los escenarios siguientes exponen sus datos de forma no deseada.

- Cuando otra persona tiene acceso al bucket de Amazon S3 al que exportó su instantánea.

Para controlar el acceso a sus instantáneas, solo permita el acceso al bucket de Amazon S3 a aquellos usuarios que desee que tengan acceso a sus datos. A fin de obtener información sobre la administración del acceso a un bucket de Amazon S3, consulte Administración del acceso en la Guía para desarrolladores de Amazon S3.

- Cuando otra persona tiene permisos para usar la operación de la API CopySnapshot.

  Los usuarios o grupos que tienen permisos para utilizar la operación de la API `CopySnapshot` pueden crear sus propios buckets de Amazon S3 y copiar las instantáneas en ellos. Para controlar el acceso a sus instantáneas, use una política (IAM) AWS Identity and Access Management para controlar quién puede usar la API de `CopySnapshot`. Para obtener más información acerca del uso de IAM para controlar el uso de las operaciones de la API de MemoryDB, consulte Administración de identidades y accesos en MemoryDB para Redis en la Guía del usuario de MemoryDB.

Temas

Paso 1: Crear un bucket de Amazon S3

El siguiente procedimiento utiliza la consola de Amazon S3 para crear un bucket de Amazon S3 al que se exporta y en el que se almacena la instantánea de MemoryDB.

Para crear un bucket de Amazon S3

1. Inicie sesión en AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Seleccione la opción Create Bucket (Crear bucket).
3. En Create a Bucket - Select a Bucket Name and Region, haga lo siguiente:

   a. En Bucket Name (Nombre del bucket), escriba un nombre para el bucket de Amazon S3.

   b. En la lista de Region (Región), elija una región de AWS para su bucket de Amazon S3. Esta región de AWS debe encontrarse en la misma región de AWS que la de la instantánea de MemoryDB que desea exportar.

c. Seleccione Crear.

Para obtener más información sobre la creación de un bucket de Amazon S3, consulte la sección de Creación de un bucket en la Guía del usuario de Amazon Simple Storage Service.

Paso 2: conceder acceso a MemoryDB a su bucket de Amazon S3

Las regiones de AWS que se presentaron antes del 20 de marzo de 2019 se encuentran habilitadas de forma predeterminada. Puede comenzar a trabajar en estas regiones de AWS de inmediato. Las regiones introducidas después del 20 de marzo de 2019 están deshabilitadas de forma predeterminada. Debe habilitar o suscribirse a estas regiones antes de poder utilizarlas, tal y como se describe en Administración de regiones de AWS.

Conceder acceso a MemoryDB a su bucket de S3 en una región de AWS

Para crear los permisos adecuados en un bucket de Amazon S3 en una región de AWS con suscripción, siga estos pasos.

Para conceder a MemoryDB acceso a un bucket de S3

1. Inicie sesión en AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Elija el nombre del bucket de Amazon S3 en el que desea copiar la instantánea. Este debe ser el bucket de S3 que creó en Paso 1: Crear un bucket de Amazon S3.
3. Elija la pestaña Permisos y en Permisos, elija Política de buckets.
4. Actualice la política para conceder a MemoryDB los permisos necesarios para realizar operaciones:

   - Agregue `[ "Service" : "region-full-name.memorydb-snapshot.amazonaws.com" ]` a `Principal`.
   - Agregue los siguientes permisos necesarios para exportar una instantánea al bucket de Amazon S3.
     - `"s3:PutObject"`
     - `"s3:GetObject"`
     - `"s3:ListBucket"`
     - `"s3:GetBucketAcl"`
     - `"s3:ListMultipartUploadParts"`

- "s3:ListBucketMultipartUploads"

A continuación, se muestra un ejemplo del aspecto que tendría la política actualizada.

```
{
    "Version": "2012-10-17",
    "Id": "Policy15397346",
    "Statement": [
        {
            "Sid": "Stmt15399483",
            "Effect": "Allow",
            "Principal": {
                "Service": "aws-region.memorydb-snapshot.amazonaws.com"
            },
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:GetBucketAcl",
                "s3:ListMultipartUploadParts",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": [
                "arn:aws:s3:::example-bucket",
                "arn:aws:s3:::example-bucket/*"
            ]
        }
    ]
}
```

Paso 3: exportar una instantánea de MemoryDB

Ya ha creado el bucket de S3 y ha concedido permisos de MemoryDB para acceder a él. Cambie la propiedad del objeto de S3 a una ACL habilitada (se prefiere el propietario del bucket). A continuación, podrá usar la consola de MemoryDB, la CLI de AWS o la API de MemoryDB para exportar su instantánea a él. En el siguiente procedimiento se da por sentado que dispone de los siguientes permisos adicionales de IAM específicos de S3.

```
{
 "Version": "2012-10-17",
```

```
 "Statement": [{
  "Effect": "Allow",
  "Action": [
   "s3:GetBucketLocation",
   "s3:ListAllMyBuckets",
   "s3:PutObject",
   "s3:GetObject",
   "s3:DeleteObject",
   "s3:ListBucket"
  ],
  "Resource": "arn:aws:s3:::*"
 }]
}
```

Exportación de una instantánea de MemoryDB (consola)

El siguiente proceso usa la consola de MemoryDB para exportar una instantánea a un bucket de Amazon S3 para que pueda tener acceso a ella desde fuera de MemoryDB. El bucket de Amazon S3 debe estar en la misma región de AWS que la instantánea de MemoryDB.

Para exportar una instantánea de MemoryDB a un bucket de Amazon S3

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. Para ver una lista de las instantáneas, en el panel de navegación izquierdo, elija Instantáneas.
3. En la lista de instantáneas, elija el botón de opción situado a la izquierda de la instantánea que desea exportar.
4. Elija Copiar.
5. En Create a Copy of the Backup? (¿Desea crear una copia del backup?), haga lo siguiente:

   a. En el cuadro Nuevo nombre de instantánea, especifique un nombre para la nueva instantánea.

      El nombre debe tener entre 1 y 1 000 caracteres y debe admitir la codificación UTF-8.

      MemoryDB agrega una partición y `.rdb` al valor que especifique aquí. Por ejemplo, si especifica `my-exported-snapshot`, MemoryDB creará `my-exported-snapshot-0001.rdb`.

   b. Desde la lista Ubicación de S3 de destino, elija el nombre del bucket de Amazon S3 al que desea copiar la instantánea (el bucket que creó en Paso 1: Crear un bucket de Amazon S3).

La ubicación de S3 de destino debe ser un bucket de Amazon S3 de la misma región de AWS que la de la instantánea con los siguientes permisos para que el proceso de exportación se realice correctamente.

- Acceso al objeto: Read (Lectura) y Write (Escritura).
- Permisos de acceso: lectura.

Para obtener más información, consulte Paso 2: conceder acceso a MemoryDB a su bucket de Amazon S3.

c. Elija Copiar.

> **Note**
>
> Si su bucket de S3 no tiene los permisos necesarios para que MemoryDB pueda exportar una instantánea, recibirá uno de los mensajes de error siguientes. Vuelva a Paso 2: conceder acceso a MemoryDB a su bucket de Amazon S3 para agregar los permisos especificados e intente de nuevo exportar la instantánea.
>
> - No se han concedido permisos de LECTURA %s a MemoryDB en el bucket de S3.
>
> Solución: añada los permisos Read en el bucket.
> - No se han concedido permisos de ESCRITURA %s a MemoryDB en el bucket de S3.
>
> Solución: añada los permisos Write en el bucket.
> - No se han concedido permisos READ_ACP a MemoryDB %s en el bucket de S3.
>
> Solución: añada Read como permiso de acceso en el bucket.

Si desea copiar su instantánea en otra región de AWS, utilice Amazon S3 para copiarla. Para obtener más información, consulte Copia de objetos en la Guía del usuario de Amazon Simple Storage Service.

Exportación de una instantánea de MemoryDB (CLI) AWS

Exporte la instantánea a un bucket de Amazon S3 con la operación de la CLI `copy-snapshot` con los siguientes parámetros:

Parameters

- `--source-snapshot-name`: nombre de la instantánea que se copiará.
- `--target-snapshot-name`: nombre de la copia de la instantánea.

  El nombre debe tener entre 1 y 1 000 caracteres y debe admitir la codificación UTF-8.

  MemoryDB agrega un identificador de partición y `.rdb` al valor que ingrese aquí. Por ejemplo, si especifica `my-exported-snapshot`, MemoryDB creará `my-exported-snapshot-0001.rdb`.
- `--target-bucket`: escriba el nombre del bucket de Amazon S3 donde desea exportar la instantánea. Se realizará una copia de la instantánea en el bucket especificado.

  El `--target-bucket` debe ser un bucket de Amazon S3 de la misma región de AWS que la de la instantánea con los siguientes permisos para que el proceso de exportación se realice correctamente.

  - Acceso al objeto: Read (Lectura) y Write (Escritura).
  - Permisos de acceso: lectura.

  Para obtener más información, consulte Paso 2: conceder acceso a MemoryDB a su bucket de Amazon S3.

La operación siguiente copia una instantánea en my-s3-bucket.

Para Linux, macOS o Unix:

```
aws memorydb copy-snapshot \
    --source-snapshot-name automatic.my-primary-2021-06-27-03-15 \
    --target-snapshot-name my-exported-snapshot \
    --target-bucket my-s3-bucket
```

Para Windows:

```
aws memorydb copy-snapshot ^
    --source-snapshot-name automatic.my-primary-2021-06-27-03-15 ^
    --target-snapshot-name my-exported-snapshot ^
    --target-bucket my-s3-bucket
```

> **Note**
>
> Si su bucket de S3 no tiene los permisos necesarios para que MemoryDB pueda exportar una instantánea, recibirá uno de los mensajes de error siguientes. Vuelva a Paso 2: conceder acceso a MemoryDB a su bucket de Amazon S3 para agregar los permisos especificados e intente de nuevo exportar la instantánea.
>
> - No se han concedido permisos de LECTURA %s a MemoryDB en el bucket de S3.
>
> Solución: añada los permisos Read en el bucket.
> - No se han concedido permisos de ESCRITURA %s a MemoryDB en el bucket de S3.
>
> Solución: añada los permisos Write en el bucket.
> - No se han concedido permisos READ_ACP a MemoryDB %s en el bucket de S3.
>
> Solución: añada Read como permiso de acceso en el bucket.

Para obtener más información, consulte `copy-snapshot` en la Referencia de los comandos de AWS CLI.

Si desea copiar su instantánea en otra región de AWS, utilice la copia de Amazon S3. Para obtener más información, consulte Copia de objetos en la Guía del usuario de Amazon Simple Storage Service.

Exportación de una instantánea de MemoryDB (API de MemoryDB)

Exporte la instantánea a un bucket de Amazon S3 con la operación de la API `CopySnapshot` con los parámetros que se indican a continuación.

Parameters

- `SourceSnapshotName`: nombre de la instantánea que se copiará.
- `TargetSnapshotName`: nombre de la copia de la instantánea.

  El nombre debe tener entre 1 y 1 000 caracteres y debe admitir la codificación UTF-8.

  MemoryDB agrega una partición y `.rdb` al valor que especifique aquí. Por ejemplo, si especifica `my-exported-snapshot`, obtendrá `my-exported-snapshot-0001.rdb`.

- `TargetBucket`: escriba el nombre del bucket de Amazon S3 donde desea exportar la instantánea. Se realizará una copia de la instantánea en el bucket especificado.

  El `TargetBucket` debe ser un bucket de Amazon S3 de la misma región de AWS que la de la instantánea con los siguientes permisos para que el proceso de exportación se realice correctamente.

  - Acceso al objeto: Read (Lectura) y Write (Escritura).
  - Permisos de acceso: lectura.

  Para obtener más información, consulte Paso 2: conceder acceso a MemoryDB a su bucket de Amazon S3.

El siguiente ejemplo hace una copia de una instantánea automática en el bucket `my-s3-bucket` de Amazon S3 .

Example

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=CopySnapshot
    &SourceSnapshotName=automatic.my-primary-2021-06-27-03-15
    &TargetBucket=my-s3-bucket
    &TargetSnapshotName=my-snapshot-copy
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210801T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

> **Note**
>
> Si su bucket de S3 no tiene los permisos necesarios para que MemoryDB pueda exportar una instantánea, recibirá uno de los mensajes de error siguientes. Vuelva a Paso 2: conceder acceso a MemoryDB a su bucket de Amazon S3 para agregar los permisos especificados e intente de nuevo exportar la instantánea.

- No se han concedido permisos de LECTURA %s a MemoryDB en el bucket de S3.

  Solución: añada los permisos Read en el bucket.
- No se han concedido permisos de ESCRITURA %s a MemoryDB en el bucket de S3.

  Solución: añada los permisos Write en el bucket.
- No se han concedido permisos READ_ACP a MemoryDB %s en el bucket de S3.

  Solución: añada Read como permiso de acceso en el bucket.

Para obtener más información, consulte CopySnapshot.

Si desea copiar su instantánea en otra región de AWS, utilice la copia de Amazon S3 para copiar la instantánea exportada al bucket de Amazon S3 de otra región de AWS. Para obtener más información, consulte Copia de objetos en la Guía del usuario de Amazon Simple Storage Service.

Restauración a partir de una instantánea

Puede restaurar los datos a partir de un archivo de instantánea .rdb de MemoryDB o ElastiCache para Redis a un nuevo clúster en cualquier momento.

El proceso de restauración de MemoryDB para Redis permite lo siguiente:

- Migración de uno o varios archivos de instantánea .rdb creados desde ElastiCache para Redis a un clúster de MemoryDB.

  Los archivos .rdb deben estar ubicados en S3 para poder realizar la restauración.
- Especificación de un número de particiones en el nuevo clúster que sea distinto del número de particiones del clúster que se usó para crear el archivo de instantánea.
- Especificación de un tipo de nodo distinto para el clúster nuevo: más grande o más pequeño. Si va a escalar a un tipo de nodo más pequeño, asegúrese de que el nuevo tipo de nodo tenga suficiente memoria para los datos y la capacidad adicional de Redis.
- Configuración de las ranuras del nuevo clúster de MemoryDB de manera distinta a la del clúster que se usó para crear el archivo de instantánea.

> **Important**
>
> - Los clústeres de MemoryDB no admiten varias bases de datos. Por tanto, al restaurar un clúster de MemoryDB se producirá un error si el archivo .rdb hace referencia a más de una base de datos.
> - No se puede restaurar una instantánea de un clúster que utiliza la organización de datos en niveles (por ejemplo, tipo de nodo r6gd) en un clúster que no utiliza la organización de datos en niveles (por ejemplo, tipo de nodo r6g).

Si realiza algún cambio al restaurar un clúster desde una instantánea, se rige por las elecciones que realice. Puede elegir estas opciones en el cuadro de diálogo Restaurar el clúster cuando utilice la consola de MemoryDB que restaurar. Para elegir las opciones, configure los valores de los parámetros cuando utilice la AWS CLI o la API de MemoryDB que restaurar.

Durante la operación de restauración, MemoryDB crea el nuevo clúster y, a continuación, lo rellena con los datos del archivo de instantánea. Cuando se complete este proceso, el clúster estará listo para aceptar solicitudes.

> **Important**
>
> Antes de continuar, asegúrese de haber creado una instantánea del clúster que desea restaurar. Para obtener más información, consulte Toma de instantáneas manuales.
> Si desea efectuar la restauración a partir de una instantánea creada externamente, consulte Inicialización de un nuevo clúster con una instantánea creada externamente.

Los procedimientos siguientes muestran cómo restaurar una instantánea a un nuevo clúster con la consola de MemoryDB, la AWS CLI o la API de MemoryDB.

Restauración a partir de una instantánea (consola)

Para restaurar una instantánea en un clúster nuevo (consola)

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación, elija Instantáneas.
3. En la lista de instantáneas, elija el botón situado junto al nombre de la instantánea desde la que desea restaurar.
4. Elija Acciones y, a continuación, Restaurar.
5. En Configuración del clúster, ingrese lo siguiente:

    a. Nombre del clúster: obligatorio. Se trata del nombre del nuevo clúster.

    b. Descripción: opcional Descripción del nuevo clúster.
6. Complete la sección Grupos de subredes:

    - En Grupos de subredes, cree un nuevo grupo de subredes o elija uno existente de la lista disponible que desee aplicar a este clúster. Si va a crear uno nuevo:

        - Escriba un nombre
        - Escriba una descripción
        - Si ha habilitado Multi-AZ, el grupo de subredes debe contener al menos dos subredes que residan en zonas de disponibilidad diferentes. Para obtener más información, consulte Subredes y grupos de subredes.

- Si va a crear un nuevo grupo de subredes y no tiene una VPC existente, se le pedirá que cree una VPC. Para obtener más información, consulte ¿Qué es Amazon VPC? en la Guía del usuario de Amazon VPC.

7. Complete la sección Configuración del clúster:

   a. Para garantizar la compatibilidad de las versiones de Redis, acepte la versión de `6.0` predeterminada.

   b. En el caso de Port, acepte 6379 como puerto predeterminado de Redis o, si tiene algún motivo para utilizar un puerto diferente, introduzca el número de puerto.

   c. En Grupo de parámetros, acepte el grupo de `default.memorydb-redis6` parámetros.

      Los grupos de parámetros controlan los parámetros de tiempo de ejecución de su clúster. Para obtener más información acerca de los grupos de parámetros, consulte Parámetros específicos de Redis.

   d. En Tipo de nodo, elija un valor para el tipo de nodo (junto con el tamaño de memoria asociado) que desee.

      Si elige un miembro de la familia de tipos de nodo r6gd, activará automáticamente la organización de datos en niveles en su clúster. Para obtener más información, consulte Organización de datos en niveles.

   e. En Número de particiones, elija el número de particiones que desea para este clúster.

      Puede cambiar dinámicamente el número de particiones del clúster. Para obtener más información, consulte Escalado de clústeres de MemoryDB.

   f. En Réplicas por partición, elija el número de nodos de réplica de lectura que desea en cada partición.

      Se aplican las siguientes restricciones.

      - Si tiene habilitado Multi-AZ, asegúrese de tener al menos una réplica por partición.
      - El número de réplicas es el mismo para cada fragmento al crear el clúster utilizando la consola.

   g. Elija Siguiente.

   h. Complete la sección Configuración avanzada:

i. En Grupos de seguridad, elija los grupos de seguridad que desea para este clúster. Un grupo de seguridad actúa como un firewall para controlar el acceso de red al clúster. Puede utilizar el grupo de seguridad predeterminado para la VPC o crear uno nuevo.

   Para obtener más información sobre los grupos de seguridad, consulte Grupos de seguridad de su VPC en la Guía del usuario de Amazon VPC.

ii. Los datos se cifran de las siguientes formas:

   - Encryption at rest (Cifrado en reposo): permite el cifrado de los datos almacenados en el disco. Para obtener más información, consulte Cifrado en reposo.

     > **Note**
     >
     > Tiene la opción de suministrar una clave de cifrado distinta al elegir Customer Managed AWS KMS key (Clave KMS de administrada por el cliente) y elegir la clave.

   - Encryption in-transit (Cifrado en tránsito): permite el cifrado de datos del cable. Esto está habilitado de forma predeterminada. Para obtener más información, consulte Cifrado en tránsito.

   Si no selecciona ningún cifrado, se creará una lista de control de acceso abierta denominada «acceso abierto» con un usuario predeterminado. Para obtener más información, consulte Autenticación de usuarios con listas de control de acceso (ACL).

iii. En el caso de una instantánea, especifique de forma opcional un periodo de retención de la instantánea y un periodo de instantáneas. De forma predeterminada, está seleccionada la opción Habilitar instantáneas automáticas.

iv. En el periodo de mantenimiento, especifique opcionalmente un periodo de mantenimiento. El periodo de mantenimiento es el tiempo, generalmente de una hora, de cada semana durante el que MemoryDB programa el mantenimiento del sistema para su clúster. Puede permitir que MemoryDB elija el día y la hora de su periodo de mantenimiento (Sin preferencia) o bien puede elegir el día, la hora y la duración por su cuenta (Especificar periodo de mantenimiento). Si elige Specify maintenance window, elija Start day, Start time y Duration (en horas) de las listas para el periodo de mantenimiento. Todas las horas se indican en UCT.

Para obtener más información, consulte Administración del mantenimiento.

v. En Notifications (Notificaciones), elija un tema existente de Amazon Simple Notification Service (Amazon SNS) o bien una entrada de ARN manual y escriba el tema nombre de recurso de Amazon (ARN). Amazon SNS le permite enviar notificaciones de inserción a dispositivos inteligentes con conexión a Internet. El valor predeterminado tiene las notificaciones deshabilitadas. Para obtener más información, consulte https://aws.amazon.com/sns/.

i. En Etiquetas, si lo desea, puede aplicar etiquetas para buscar y filtrar sus clústeres o realizar un seguimiento de sus costos de AWS.

j. Revise todas las entradas y opciones y, a continuación, realice todos los cambios necesarios. Cuando esté listo, elija Create cluster para lanzar su clúster, o bien Cancel para cancelar la operación.

En cuanto el estado de su clúster sea available, podrá conceder a EC2 acceso a este, así como conectarse a él y comenzar a utilizarlo. Para más información, consulte Paso 2: autorizar el acceso al clúster y Paso 3: conectar al clúster.

> **Important**
>
> Cuando su clúster esté disponible, se cobrará por cada hora u hora parcial que el clúster esté activo, incluso si no lo está utilizando de forma activa. Para dejar de incurrir en cargos por este clúster, debe eliminarlo. Consulte Paso 4: eliminar un clúster.

Restauración a partir de una instantánea (CLI de AWS)

Cuando use la operación `create-cluster`, asegúrese de incluir el parámetro `--snapshot-name` o `--snapshot-arns` para inicializar el nuevo clúster con los datos de la instantánea.

Para obtener más información, consulte los siguientes temas:

- Creación de un clúster (AWS CLI) en la Guía del usuario de MemoryDB.
- create-cluster en la referencia de comandos de la AWS CLI.

Restauración a partir de una instantánea (API de MemoryDB)

Puede restaurar una instantánea de MemoryDB mediante la operación de la API de MemoryDB `CreateCluster`.

Cuando use la operación `CreateCluster`, asegúrese de incluir el parámetro `SnapshotName` o `SnapshotArns` para inicializar el nuevo clúster con los datos de la instantánea.

Para obtener más información, consulte los siguientes temas:

- Creación de un clúster (API de MemoryDB) en la Guía del usuario de MemoryDB.
- CreateCluster en la Referencia de la API de MemoryDB.

Inicialización de un nuevo clúster con una instantánea creada externamente

Cuando se crea un nuevo clúster de MemoryDB, puede inicializarlo con los datos de un archivo de instantánea .rdb de Redis.

Para iniciar un nuevo clúster de MemoryDB a partir de una instantánea de MemoryDB o una instantánea de ElastiCache para Redis, consulte Restauración a partir de una instantánea.

Cuando use un archivo .rdb de Redis para propagar datos a un nuevo clúster de MemoryDB, podrá realizar lo siguiente:

- Especifique el número de particiones del nuevo clúster. Este número puede ser distinto del número de particiones del clúster que se utilizó para crear el archivo de instantánea.
- Especificar un tipo de nodo distinto para el nuevo clúster, más grande o más pequeño que el que se utilizó en el clúster que creó la instantánea. Si escala a un tipo de nodo más pequeño, asegúrese de que el nuevo tipo de nodo tenga suficiente memoria para los datos y la capacidad adicional de Redis.

Important

- Debe asegurarse de que los datos de la instantánea no superen los recursos del nodo.

  Si la instantánea es demasiado grande, el clúster resultante tendrá el estado `restore-failed`. Si esto ocurre, deberá eliminar el clúster y empezar de nuevo.

  Para ver una lista completa de los distintos tipos de nodos y las especificaciones, consulte Parámetros específicos de tipo de nodo de MemoryDB.
- Solo puede cifrar un archivo .rdb de Redis con cifrado del lado del servidor de Amazon S3 (SSE-S3). Para obtener más información, consulte Protección de los datos con el cifrado del lado del servidor.

Paso 1: crear una instantánea de Redis en un clúster externo

Para crear la instantánea para iniciar su clúster de MemoryDB

1. Conéctese a su instancia de Redis existente.

2. Ejecute la operación de Redis `BGSAVE` o `SAVE` para crear una instantánea. Tenga en cuenta la ubicación de su archivo .rdb.

   `BGSAVE` es una operación asincrónica y no bloquea otros clientes durante el procesamiento. Para obtener más información, consulte la operación BGSAVE en el sitio web de Redis.

   `SAVE` es una operación sincrónica y bloquea otros procesos hasta que finalice. Para obtener más información, consulte la operación SAVE en el sitio web de Redis.

Para obtener información adicional sobre la creación de instantáneas, consulte Persistencia de Redis en el sitio web de Redis.

Paso 2: crear un bucket y una carpeta de Amazon S3

Una vez que se crea el archivo de instantánea, deberá cargarlo en una carpeta de un bucket de Amazon S3. Para ello, primero debe disponer de un bucket de Amazon S3 y de una carpeta en dicho bucket. Si ya dispone de un bucket de Amazon S3 y una carpeta con los permisos pertinentes, puede pasar a Paso 3: cargar la instantánea a Amazon S3.

Para crear un bucket de Amazon S3

1. Inicie sesión en AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Siga las instrucciones para crear un bucket de Amazon S3 en Creación de un bucket en la Guía del usuario de Amazon Simple Storage Service.

   El nombre del bucket de Amazon S3 debe estar conforme con DNS. De lo contrario, MemoryDB no podrá acceder al archivo de copia de seguridad. Las reglas para la conformidad con DNS son:

   - Los nombres deben tener un mínimo de 3 y un máximo de 63 caracteres de largo.
   - Los nombres deben ser una serie de una o más etiquetas separadas por un punto (.) en el que cada etiqueta:
     - Comienza por una letra minúscula o un número.
     - Termina con una letra minúscula o un número.
     - Solo contiene letras minúsculas, números y guiones.
   - Los nombres no pueden tener el formato de una dirección IP (por ejemplo, 192.0.2.0).

Recomendamos encarecidamente que cree su bucket de Amazon S3 en la misma región de AWS que la del nuevo clúster de MemoryDB. Este enfoque garantiza la mayor velocidad de transferencia de datos posible cuando MemoryDB lea el archivo .rdb desde Amazon S3.

> **Note**
>
> Para conservar la máxima seguridad de los datos, asegúrese de que los permisos de su bucket de Amazon S3 sean lo más restrictivos posible. Al mismo tiempo, los permisos seguirán necesitando permitir que se utilicen el bucket y su contenido para generar su nuevo clúster de MemoryDB.

Para agregar una carpeta a un bucket de Amazon S3

1. Inicie sesión en AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Elija el nombre del bucket en el que va a cargar el archivo .rdb.
3. Elija Crear carpeta.
4. Escriba un nombre para la nueva carpeta.
5. Elija Guardar.

   Anote el nombre del bucket y el nombre de la carpeta.

Paso 3: cargar la instantánea a Amazon S3

Ahora, cargue el archivo .rdb que creó en Paso 1: crear una instantánea de Redis en un clúster externo. Cárguelo en el bucket de Amazon S3 y la carpeta que creó en Paso 2: crear un bucket y una carpeta de Amazon S3. Para obtener más información acerca de esta tarea, consulte Carga de objetos. Entre los pasos 2 y 3, elija el nombre de la carpeta que creó.

Para cargar el archivo .rdb a una carpeta de Amazon S3

1. Inicie sesión en AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Elija el nombre del bucket de Amazon S3 que creó en el paso 2.
3. Elija el nombre de la carpeta que creó en el paso 2.

4. Seleccione Cargar.
5. Elija Add files.
6. Examine el archivo o los archivos que desea cargar y, a continuación, elija el archivo o los archivos. Para elegir varios archivos, mantenga pulsada la tecla Ctrl al mismo tiempo que selecciona un nombre de archivo.
7. Elija Open.
8. Asegúrese de que se muestran los archivos correctos en la página Cargar y, a continuación, elija Cargar.

Escriba la ruta del archivo .rdb. Por ejemplo, si el nombre del bucket es `myBucket` y la ruta es `myFolder/redis.rdb`, escriba `myBucket/myFolder/redis.rdb`. Necesitará esta ruta para propagar en el nuevo clúster los datos de la instantánea.

Para obtener información adicional, consulte Reglas de nomenclatura de buckets en la Guía del usuario de Amazon Simple Storage Service.

Paso 4: conceder a MemoryDB acceso de lectura al archivo .rdb

Las regiones de AWS que se presentaron antes del 20 de marzo de 2019 se encuentran habilitadas de forma predeterminada. Puede comenzar a trabajar en estas regiones de AWS de inmediato. Las regiones introducidas después del 20 de marzo de 2019 están deshabilitadas de forma predeterminada. Debe habilitar o suscribirse a estas regiones antes de poder utilizarlas, tal y como se describe en Administración de regiones de AWS.

Concesión a MemoryDB de acceso de lectura al archivo .rdb

Para conceder a MemoryDB acceso de lectura al archivo de instantánea

1. Inicie sesión en AWS Management Console Management Console y abra la consola de Amazon S3 en https://console.aws.amazon.com/s3/.
2. Elija el nombre del bucket de S3 que contiene su archivo .rdb.
3. Elija el nombre de la carpeta que contiene su archivo .rdb.
4. Elija el nombre de su archivo de instantánea .rdb. El nombre del archivo seleccionado aparecerá encima de las pestañas, en la parte superior de la página.
5. Elija la pestaña Permisos.
6. En Permissions (Permisos), elija Bucket policy (Política de bucket), y luego Edit (Editar).

7. Actualice la política para conceder a MemoryDB los permisos necesarios para realizar operaciones:

   - Agregue `[ "Service" : "region-full-name.memorydb-snapshot.amazonaws.com" ]` a `Principal`.
   - Agregue los siguientes permisos necesarios para exportar una instantánea al bucket de Amazon S3:
     - `"s3:GetObject"`
     - `"s3:ListBucket"`
     - `"s3:GetBucketAcl"`

   A continuación, se muestra un ejemplo del aspecto que tendría la política actualizada.

```
{
    "Version": "2012-10-17",
    "Id": "Policy15397346",
    "Statement": [
        {
            "Sid": "Stmt15399483",
            "Effect": "Allow",
            "Principal": {
                "Service": "us-east-1.memorydb-snapshot.amazonaws.com"
            },
            "Action": [
                "s3:GetObject",
                "s3:ListBucket",
                "s3:GetBucketAcl"
            ],
            "Resource": [
                "arn:aws:s3:::example-bucket",
                "arn:aws:s3:::example-bucket/snapshot1.rdb",
                "arn:aws:s3:::example-bucket/snapshot2.rdb"
            ]
        }
    ]
}
```

8. Elija Guardar.

Paso 5: inicialización del clúster de MemoryDB con los datos del archivo .rdb

Ahora está listo para crear un clúster de MemoryDB y propagar los datos del archivo .rdb. Para crear el clúster, siga las instrucciones que se detallan en Creación de un clúster de MemoryDB.

El método que utilice para indicar a MemoryDB dónde buscar la instantánea de Redis que cargó en Amazon S3 dependerá del método que utilice para crear el clúster:

Inicialización del clúster de MemoryDB con los datos del archivo .rdb

- Uso de la consola de MemoryDB

  Tras elegir el motor de Redis, expanda la sección Advanced Redis settings y busque la opción Import data to cluster. En el cuadro Seed RDB file S3 location (Inicializar ubicación de S3 del archivo RDB), escriba la ruta de Amazon S3 de los archivos. Si tiene varios archivos.rdb, escriba la ruta para cada archivo en una lista separada por comas. La ruta de Amazon S3 tendrá un aspecto similar a *myBucket*/*myFolder*/*myBackupFilename*.rdb.
- Uso de AWS CLI

  Si usa la operación `create-cluster` o `create-cluster`, use el parámetro `--snapshot-arns` para especificar un ARN completo para cada archivo .rdb. Por ejemplo, `arn:aws:s3:::`*myBucket*/*myFolder*/*myBackupFilename*`.rdb`. El ARN debe resolverse en los archivos de instantánea que almacenó en Amazon S3.
- Uso de la API de MemoryDB

  Si usa las operaciones `CreateCluster` o `CreateCluster` de la API de MemoryDB, use el parámetro `SnapshotArns` para especificar un ARN completo para cada archivo .rdb. Por ejemplo, `arn:aws:s3:::`*myBucket*/*myFolder*/*myBackupFilename*`.rdb`. El ARN debe resolverse en los archivos de instantánea que almacenó en Amazon S3.

Durante el proceso de creación del clúster, los datos de su instantánea se escribirán en el clúster. Puede monitorear el progreso consultando los mensajes de eventos de MemoryDB. Para ello, vaya a la consola de MemoryDB y elija Eventos. También puede usar la interfaz de línea de comandos de AWS MemoryDB o la API de MemoryDB para obtener los mensajes de eventos.

Etiquetado de instantáneas

Puede asignar sus propios metadatos a cada instantánea en forma de etiquetas. Las etiquetas permiten clasificar las instantáneas de diversas maneras, por ejemplo, según su finalidad, propietario o entorno. Esto es útil cuando tiene muchos recursos del mismo tipo: puede identificar rápidamente un recurso específico en función de las etiquetas que le haya asignado. Para obtener más información, consulte Recursos que se pueden etiquetar.

Las etiquetas de asignación de costos permiten realizar un seguimiento de los costos de varios servicios de AWS al agrupar los gastos de las facturas por valores de etiqueta. Para obtener más información sobre las etiquetas de asignación de costos, consulte Uso de etiquetas de asignación de costos.

Use la consola de MemoryDB, la AWS CLI o la API de MemoryDB para agregar, mostrar, modificar, eliminar o copiar las etiquetas de asignación de costos en sus instantáneas. Para obtener más información, consulte Monitoreo de costos con etiquetas de asignación de costos.

Eliminación de una instantánea

Las instantáneas automáticas se eliminan automáticamente cuando finaliza el límite de retención. Si elimina un clúster, también se eliminarán todas sus instantáneas automáticas.

MemoryDB ofrece una operación de la API de eliminación que permite eliminar instantáneas en cualquier momento, independientemente de si la instantánea se creó de forma automática o manual. Dado que las instantáneas manuales no tienen límite de retención, estas copias solo se pueden eliminar de forma manual.

Puede eliminar una instantánea con la consola de MemoryDB, la AWS CLI o la API de MemoryDB.

Eliminación de una instantánea (consola)

El siguiente procedimiento elimina una instantánea mediante la consola de MemoryDB.

Eliminar una instantánea

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación de la izquierda, elija Instantáneas.

   Aparece la pantalla Instantáneas con una lista de sus instantáneas.
3. Elija el botón de opción situado a la izquierda del nombre de la instantánea que desee eliminar.
4. Elija Acciones y, a continuación, elija Eliminar.
5. Si desea eliminar esta instantánea, introduzca `delete` en el cuadro de texto y, a continuación, seleccione Eliminar. Para cancelar la eliminación, elija Cancelar. El estado cambia a deleting.

Eliminación de una instantánea (CLI de AWS)

Use la operación de la AWS CLI delete-snapshot con el parámetro siguiente para eliminar una instantánea.

- `--snapshot-name`: nombre de la instantánea que se va a eliminar.

El código siguiente elimina la instantánea `myBackup`.

```
aws memorydb delete-snapshot --snapshot-name myBackup
```

Para obtener más información, consulte delete-snapshot en la Referencia de comandos de la AWS CLI.

Eliminar una instantánea (API de MemoryDB)

Use la operación de la API `DeleteSnapshot` con el parámetro siguiente para eliminar una instantánea.

- `SnapshotName`: nombre de la instantánea que se va a eliminar.

El código siguiente elimina la instantánea `myBackup`.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DeleteSnapshot
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &SnapshotName=myBackup
   &Timestamp=20210802T192317Z
   &Version=2021-01-01
   &X-Amz-Credential=<credential>
```

Para obtener más información, consulte DeleteSnapshot.

Escalado

La cantidad de datos que necesita su aplicación para procesar casi nunca es fija. Aumenta y disminuye a medida que su negocio crece o experimenta las fluctuaciones normales de la demanda. Si administra por sí mismo sus aplicaciones, necesita aprovisionar suficiente hardware para los picos de demanda, lo cual puede resultar caro. Al utilizar MemoryDB para Redis, puede escalar para satisfacer la demanda actual, pagando solo por lo que utilice.

Lo siguiente lo ayuda a encontrar el tema correcto para las acciones de escalado que desea realizar.

Escalar MemoryDB

Acción	MemoryDB	
Escalado ascendente	Cambios en las particiones y reequilibrado de particiones en línea para MemoryDB	

Acción	MemoryDB	
Cambios de tipos de nodos	Escalado vertical en línea mediante la modificación del tipo de nodo	
Cambio del número de particiones	Escalado de clústeres de MemoryDB	

Escalado de clústeres de MemoryDB

A medida que cambie la demanda en los clústeres, puede decidir mejorar el desempeño o reducir los costos cambiando el número de particiones en su clúster de MemoryDB. Recomendamos que utilice el escalado horizontal online, ya que permite que el clúster continúe sirviendo las solicitudes durante el proceso de escalado.

Entre las condiciones en las que puede decidir cambiar el escalado de su clúster se incluyen las siguientes:

- Presión de memoria:

  Si los nodos del clúster tienen presión de la memoria, puede decidir realizar un escalado ascendente para tener más recursos con el fin de almacenar los datos y servir las solicitudes mejor.

  Puede determinar si los nodos se encuentran bajo presión de memoria monitoreando las siguientes métricas: FreeableMemory, SwapUsage y BytesUsedForMemoryDB.
- Cuello de botella de CPU o de red:

  Si se producen muchos problemas de latencia o rendimiento en su clúster, tal vez deba hacer un escalado ascendente para resolverlos.

  Puede monitorear los niveles de latencia y rendimiento mediante el monitoreo de las siguientes métricas: CPUUtilization, NetworkBytesIn, NetworkBytesOut, CurrConnections y NewConnections.
- El escalado del clúster es excesivo:

  La demanda actual en su clúster es tal que el escalado descendente no afecta al rendimiento y reduce los costos.

  Puede monitorear el uso del clúster para determinar si puede realizar una reducción horizontal con las siguientes métricas: FreeableMemory, SwapUsage, BytesUsedForMemoryDB, CPUUtilization, NetworkBytesIn, NetworkBytesOut, CurrConnections y NewConnections.

Impacto de rendimiento del escalado

Cuando escala utilizando el proceso sin conexión, el clúster no está en línea durante una parte importante del proceso y, por tanto, no puede atender las solicitudes. Cuando escala utilizando el método online, como el escalado es una operación que realiza un uso intensivo de computación, se

deteriora algo el rendimiento, aunque el clúster sigue atendiendo las solicitudes en toda la operación de escalado. El nivel de deterioro de la experiencia depende del uso normal de la CPU y sus datos.

Existen dos formas de escalar el clúster de MemoryDB: el escalado horizontal y vertical.

- El escalado horizontal le permite cambiar el número de particiones del clúster agregando o eliminando particiones. El proceso de partición en línea le permite escalar verticalmente/horizontalmente mientras el clúster sigue ofreciendo solicitudes entrantes.
- Escalado vertical: cambie el tipo de nodo para cambiar el tamaño del clúster. El proceso de escalado vertical online le permite el escalado ascendente y descendente mientras el clúster sigue ofreciendo solicitudes entrantes.

Si reduce la el tamaño y la capacidad de memoria del clúster mediante el escalado descendente, asegúrese de que la nueva configuración disponga de memoria suficiente para sus datos y la capacidad adicional de Redis.

Cambios en las particiones y reequilibrado de particiones sin conexión para MemoryDB

La ventaja principal que obtiene de la reconfiguración de particiones sin conexión es que puede hacer algo más que agregar o eliminar particiones de su clúster. Al hacer cambios de las particiones sin conexión, además de cambiar el número de particiones del clúster, puede hacer lo siguiente:

- Cambia el tipo de nodo de su clúster.
- Actualizar a una nueva versión del motor.

> **Note**
>
> Los cambios de particiones sin conexión no se admiten en los clústeres con la organización de datos en niveles habilitada. Para obtener más información, consulte Organización de datos en niveles..

La desventaja principal de la reconfiguración de particiones sin conexión es que el clúster está sin conexión al comentar la parte de restauración del proceso y así continuará hasta que actualice los puntos de conexión de la aplicación. El tiempo que el clúster está sin conexión varía según la cantidad de datos del clúster.

Para reconfigurar las particiones del clúster de MemoryDB sin conexión

1. Cree una instantánea manual de su clúster de MemoryDB existente. Para obtener más información, consulte Toma de instantáneas manuales.
2. Cree un nuevo clúster restaurándolo a partir de la instantánea. Para obtener más información, consulte Restauración a partir de una instantánea.
3. Actualice los puntos de conexión de la aplicación a los puntos de conexión del nuevo clúster. Para obtener más información, consulte Búsqueda de puntos de conexión.

Cambios en las particiones y reequilibrado de particiones en línea para MemoryDB

Mediante el uso del cambio y el reequilibrio de las particiones en línea con MemoryDB, puede escalar su MemoryDB dinámicamente sin tiempo de inactividad. Este enfoque significa que el clúster puede seguir atendiendo las solicitudes, incluso mientras esté en curso el escalado o el reequilibrado.

Puede hacer lo siguiente:

- Escalar horizontalmente: aumente la capacidad de lectura y escritura añadiendo particiones a su clúster de MemoryDB.

  Si agrega uno o varias particiones a su clúster, el número de nodos de cada nueva partición es el mismo que el número de nodos en el menor de las particiones existentes.
- Reducción horizontal: reduzca la capacidad de lectura y escritura, y, por lo tanto, los costos, eliminando particiones del clúster de MemoryDB.

En la actualidad, las siguientes limitaciones se aplican a los cambios de particiones en línea de MemoryDB:

- Existen limitaciones con ranuras o espacios de claves y grandes elementos:

  Si alguna de las claves de una partición contiene un elemento grande, la clave no se puede migrar a una partición nueva en el escalado ascendente o el reequilibrado. Esta funcionalidad puede provocar particiones desequilibradas.

  Si alguna de las claves de una partición contiene un elemento grande (elementos mayores que 256 MB después de la serialización), esa partición no se elimina en la reducción horizontal. Esta funcionalidad puede provocar que algunas particiones no se eliminen.

- Al realizar el escalado horizontal, el número de nodos de cualquier partición nueva es igual al número de nodo de la partición existente.

Para obtener más información, consulte Prácticas recomendadas: redimensionamiento de clústeres en línea.

Puede escalar horizontalmente o reequilibrar sus clústeres de MemoryDB mediante la AWS Management Console, la AWS CLI y la API de MemoryDB.

Adición de particiones con los cambios de particiones en línea

Puede añadir particiones a su clúster de MemoryDB con la AWS Management Console, la AWS CLI o la API de MemoryDB.

Adición de particiones (consola)

Puede usar la AWS Management Console para agregar una o varias particiones a su clúster de MemoryDB. El siguiente procedimiento describe el proceso.

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. En la lista de clústeres, elija el nombre del clúster del que desea agregar una partición.
3. En la pestaña Particiones y nodos, seleccione Agregar o eliminar particiones
4. En Nuevo número de particiones, introduzca el número de particiones que desee.
5. Seleccione Confirmar para conservar los cambios o Cancelar para descartarlos.

Adición de particiones (AWS CLI)

En el siguiente proceso se describe cómo reconfigurar las particiones de su clúster de MemoryDB añadiendo particiones mediante la AWS CLI.

Use los siguientes parámetros con `update-cluster`.

Parámetros

- `--cluster-name`: obligatorio. Especifica en qué clúster (clúster) se debe realizar la operación de reconfiguración de particiones.
- `--shard-configuration`: obligatorio. Le permite establecer el número de particiones.

- `ShardCount`: defina esta propiedad para especificar el número de particiones que desea.

Example

En el siguiente ejemplo, se modifica el número de particiones del clúster `my-cluster` a 2.

Para Linux, macOS o Unix:

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --shard-configuration \
        ShardCount=2
```

Para Windows:

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
    --shard-configuration ^
        ShardCount=2
```

Devuelve la siguiente respuesta JSON:

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "updating",
        "NumberOfShards": 2,
        "AvailabilityMode": "MultiAZ",
        "ClusterEndpoint": {
            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-east-1.amazonaws.com",
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
```

```
        "SnapshotWindow": "04:30-05:30",
        "DataTiering": "false",
        "AutoMinorVersionUpgrade": true
    }
}
```

Para ver los detalles del clúster actualizado una vez que su estado cambie de actualizado a disponible, utilice el siguiente comando:

Para Linux, macOS o Unix:

```
aws memorydb describe-clusters \
    --cluster-name my-cluster
    --show-shard-details
```

Para Windows:

```
aws memorydb describe-clusters ^
    --cluster-name my-cluster
    --show-shard-details
```

Devuelve la siguiente respuesta JSON:

```
{
    "Clusters": [
        {
            "Name": "my-cluster",
            "Status": "available",
            "NumberOfShards": 2,
            "Shards": [
                {
                    "Name": "0001",
                    "Status": "available",
                    "Slots": "0-8191",
                    "Nodes": [
                        {
                            "Name": "my-cluster-0001-001",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
```

```
                            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                            "Port": 6379
                        }
                    },
                    {
                        "Name": "my-cluster-0001-002",
                        "Status": "available",
                        "AvailabilityZone": "us-east-1b",
                        "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                        "Endpoint": {
                            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                            "Port": 6379
                        }
                    }
                ],
                "NumberOfNodes": 2
            },
            {
                "Name": "0002",
                "Status": "available",
                "Slots": "8192-16383",
                "Nodes": [
                    {
                        "Name": "my-cluster-0002-001",
                        "Status": "available",
                        "AvailabilityZone": "us-east-1b",
                        "CreateTime": "2021-08-22T14:26:18.693000-07:00",
                        "Endpoint": {
                            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                            "Port": 6379
                        }
                    },
                    {
                        "Name": "my-cluster-0002-002",
                        "Status": "available",
                        "AvailabilityZone": "us-east-1a",
                        "CreateTime": "2021-08-22T14:26:18.765000-07:00",
                        "Endpoint": {
                            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                            "Port": 6379
```

```
                            }
                        }
                    ],
                    "NumberOfNodes": 2
                }
            ],
            "ClusterEndpoint": {
                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                "Port": 6379
            },
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "ParameterGroupName": "default.memorydb-redis6",
            "ParameterGroupStatus": "in-sync",
            "SubnetGroupName": "my-sg",
            "TLSEnabled": true,
            "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
            "SnapshotRetentionLimit": 0,
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "SnapshotWindow": "04:30-05:30",
            "ACLName": "my-acl",
            "DataTiering": "false",
            "AutoMinorVersionUpgrade": true
        }
    ]
}
```

Para obtener más información, consulte update-cluster en la Referencia de comandos de la AWS CLI.

Adición de particiones (API de MemoryDB)

Puede usar la API de MemoryDB para reconfigurar las particiones de su clúster de MemoryDB online mediante la operación `UpdateCluster`.

Use los siguientes parámetros con `UpdateCluster`.

Parámetros

- `ClusterName`: obligatorio. Especifica en qué clúster se debe realizar la operación de reconfiguración de particiones.

- `ShardConfiguration`: obligatorio. Le permite establecer el número de particiones.
  - `ShardCount`: defina esta propiedad para especificar el número de particiones que desea.

Para obtener más información, consulte UpdateCluster.

Eliminación de particiones con los cambios de particiones en línea

Puede eliminar particiones de su clúster de MemoryDB con la AWS Management Console, la AWS CLI o la API de MemoryDB.

Eliminación de particiones (consola)

En el siguiente proceso se describe cómo reconfigurar las particiones de su clúster de MemoryDB eliminando particiones mediante la AWS Management Console.

> **Important**
>
> Antes de eliminar particiones de su clúster, MemoryDB comprueba que todos los datos van a caber en las particiones restantes. Si los datos caben, las particiones se eliminan del clúster según lo solicitado. Si los datos no van a caber en las particiones restantes, el proceso se termina y el clúster se deja con la misma configuración de partición de antes de que se hiciera la solicitud.

Puede usar la AWS Management Console para eliminar una o varias particiones de su clúster de MemoryDB. No puede eliminar todas las particiones de un clúster. En su lugar, debe eliminar el clúster. Para obtener más información, consulte Paso 4: eliminar un clúster. El siguiente procedimiento describe el proceso para eliminar una o varias particiones.

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. En la lista de clústeres, elija el nombre del clúster del que desea quitar una partición.
3. En la pestaña Particiones y nodos, seleccione Agregar o eliminar particiones
4. En Nuevo número de particiones, introduzca el número de particiones que desee (con un mínimo de 1).
5. Seleccione Confirmar para conservar los cambios o Cancelar para descartarlos.

Eliminación de particiones (AWS CLI)

En el siguiente proceso se describe cómo reconfigurar las particiones de su clúster de MemoryDB eliminando particiones mediante la AWS CLI.

> **Important**
>
> Antes de eliminar particiones de su clúster, MemoryDB comprueba que todos los datos van a caber en las particiones restantes. Si los datos caben, las particiones especificadas se eliminan del clúster según lo solicitado y sus espacios de claves se asignan a las particiones restantes. Si los datos no van a caber en las particiones restantes, el proceso se termina y el clúster se deja con la misma configuración de partición de antes de que se hiciera la solicitud.

Puede usar la AWS CLI para eliminar una o varias particiones de su clúster de MemoryDB. No puede eliminar todas las particiones de un clúster. En su lugar, debe eliminar el clúster. Para obtener más información, consulte Paso 4: eliminar un clúster.

Use los siguientes parámetros con `update-cluster`.

Parámetros

- `--cluster-name`: obligatorio. Especifica en qué clúster (clúster) se debe realizar la operación de reconfiguración de particiones.
- `--shard-configuration`: obligatorio. Le permite establecer el número de particiones mediante la propiedad `ShardCount`:

  `ShardCount`: defina esta propiedad para especificar el número de particiones que desea.

Example

En el siguiente ejemplo, se modifica el número de particiones del clúster `my-cluster` a 2.

Para Linux, macOS o Unix:

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --shard-configuration \
        ShardCount=2
```

Para Windows:

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
    --shard-configuration ^
        ShardCount=2
```

Devuelve la siguiente respuesta JSON:

```
{
    "Cluster": {
        "Name": "my-cluster",
        "Status": "updating",
        "NumberOfShards": 2,
        "AvailabilityMode": "MultiAZ",
        "ClusterEndpoint": {
            "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-east-1.amazonaws.com",
            "Port": 6379
        },
        "NodeType": "db.r6g.large",
        "EngineVersion": "6.2",
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupName": "default.memorydb-redis6",
        "ParameterGroupStatus": "in-sync",
        "SubnetGroupName": "my-sg",
        "TLSEnabled": true,
        "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
        "SnapshotRetentionLimit": 0,
        "MaintenanceWindow": "wed:03:00-wed:04:00",
        "SnapshotWindow": "04:30-05:30",
        "DataTiering": "false",
        "AutoMinorVersionUpgrade": true
    }
}
```

Para ver los detalles del clúster actualizado una vez que su estado cambie de actualizado a disponible, utilice el siguiente comando:

Para Linux, macOS o Unix:

```
aws memorydb describe-clusters \
    --cluster-name my-cluster
```

```
    --show-shard-details
```

Para Windows:

```
aws memorydb describe-clusters ^
    --cluster-name my-cluster
    --show-shard-details
```

Devuelve la siguiente respuesta JSON:

```
{
    "Clusters": [
        {
            "Name": "my-cluster",
            "Status": "available",
            "NumberOfShards": 2,
            "Shards": [
                {
                    "Name": "0001",
                    "Status": "available",
                    "Slots": "0-8191",
                    "Nodes": [
                        {
                            "Name": "my-cluster-0001-001",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        },
                        {
                            "Name": "my-cluster-0001-002",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1b",
                            "CreateTime": "2021-08-21T20:22:12.405000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
```

```
                            }
                        }
                    ],
                    "NumberOfNodes": 2
                },
                {
                    "Name": "0002",
                    "Status": "available",
                    "Slots": "8192-16383",
                    "Nodes": [
                        {
                            "Name": "my-cluster-0002-001",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1b",
                            "CreateTime": "2021-08-22T14:26:18.693000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        },
                        {
                            "Name": "my-cluster-0002-002",
                            "Status": "available",
                            "AvailabilityZone": "us-east-1a",
                            "CreateTime": "2021-08-22T14:26:18.765000-07:00",
                            "Endpoint": {
                                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                                "Port": 6379
                            }
                        }
                    ],
                    "NumberOfNodes": 2
                }
            ],
            "ClusterEndpoint": {
                "Address": "clustercfg.my-cluster.xxxxxx.memorydb.us-
east-1.amazonaws.com",
                "Port": 6379
            },
            "NodeType": "db.r6g.large",
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
```

```
            "ParameterGroupName": "default.memorydb-redis6",
            "ParameterGroupStatus": "in-sync",
            "SubnetGroupName": "my-sg",
            "TLSEnabled": true,
            "ARN": "arn:aws:memorydb:us-east-1:xxxxxxexamplearn:cluster/my-cluster",
            "SnapshotRetentionLimit": 0,
            "MaintenanceWindow": "wed:03:00-wed:04:00",
            "SnapshotWindow": "04:30-05:30",
            "ACLName": "my-acl",
            "DataTiering": "false",
            "AutoMinorVersionUpgrade": true
        }
    ]
}
```

Para obtener más información, consulte update-cluster en la Referencia de comandos de la AWS CLI.

Eliminación de particiones (API de MemoryDB)

Puede usar la API de MemoryDB para reconfigurar las particiones de su clúster de MemoryDB online mediante la operación `UpdateCluster`.

En el siguiente proceso se describe cómo reconfigurar las particiones de su clúster de MemoryDB eliminando particiones mediante la API de MemoryDB.

> **Important**
>
> Antes de eliminar particiones de su clúster, MemoryDB comprueba que todos los datos van a caber en las particiones restantes. Si los datos caben, las particiones especificadas se eliminan del clúster según lo solicitado y sus espacios de claves se asignan a las particiones restantes. Si los datos no van a caber en las particiones restantes, el proceso se termina y el clúster se deja con la misma configuración de partición de antes de que se hiciera la solicitud.

Puede usar la API de MemoryDB para eliminar una o varias particiones de su clúster de MemoryDB. No puede eliminar todas las particiones de un clúster. En su lugar, debe eliminar el clúster. Para obtener más información, consulte Paso 4: eliminar un clúster.

Use los siguientes parámetros con `UpdateCluster`.

Parámetros

- `ClusterName`: obligatorio. Especifica en qué clúster (clúster) se debe realizar la operación de reconfiguración de particiones.
- `ShardConfiguration`: obligatorio. Le permite establecer el número de particiones mediante la propiedad `ShardCount`:

  `ShardCount`: defina esta propiedad para especificar el número de particiones que desea.

Escalado vertical en línea mediante la modificación del tipo de nodo

Mediante el escalado vertical en línea con MemoryDB, puede escalar el clúster dinámicamente con un tiempo de inactividad mínimo. Esto permite que el clúster atienda solicitudes incluso mientras se escala.

> **Note**
>
> No se admite el escalado entre un clúster de organización de datos en niveles (por ejemplo, un clúster que utiliza un tipo de nodo r6gd) y un clúster que no utiliza la organización de datos en niveles (por ejemplo, un clúster que utiliza un tipo de nodo r6g). Para obtener más información, consulte Organización de datos en niveles.

Puede hacer lo siguiente:

- Escalado vertical: aumente la capacidad de lectura y escritura ajustando el tipo de nodo del clúster de MemoryDB para utilizar un tipo de nodo más grande.

  MemoryDB redimensiona dinámicamente su clúster mientras permanece en línea y atiende solicitudes.
- Reducción vertical: reduzca verticalmente la capacidad de lectura y escritura al ajustar el tipo de nodo para utilizar un nodo más pequeño. Nuevamente, MemoryDB redimensiona dinámicamente su clúster mientras permanece en línea y atiende solicitudes. En este caso, reduzca los costos reduciendo el tamaño del nodo.

> **Note**
>
> Los procesos de escalado ascendente y descendente dependen de la creación de clústeres con tipos de nodo seleccionados recientemente y la sincronización de los nuevos nodos con los anteriores. Para garantizar un flujo de escalado ascendente/descendente uniforme, realice el siguiente procedimiento:
>
> - Aunque el proceso de escalado vertical está diseñado para que sea completamente online, se basa en la sincronización de datos entre el nodo antiguo y el nuevo. Recomendamos que inicie el escalado ascendente/descendente durante las horas en las que espera que el tráfico de datos sea mínimo.
> - Pruebe el comportamiento de la aplicación durante el escalado en un entorno de ensayo, si es posible.

Escalado vertical en línea

Temas

- Escalado vertical de clústeres de MemoryDB (consola)
- Ampliación de clústeres de MemoryDB (CLI) AWS
- Ampliación de clústeres de MemoryDB (API de MemoryDB)

Escalado vertical de clústeres de MemoryDB (consola)

El siguiente procedimiento describe cómo escalar verticalmente un clúster de MemoryDB mediante la AWS Management Console. Durante este proceso, el clúster de MemoryDB seguirá atendiendo solicitudes con un tiempo de inactividad mínimo.

Para escalar verticalmente un clúster (consola)

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. En la lista de clústeres, elija el clúster.
3. Elija Actions (Acciones) y después Modify (Modificar).
4. En el cuadro de diálogo Modificar clúster:

- Elija el tipo de nodo que desee ampliar en la lista Node type. Para aplicar el escalado ascendente, seleccione un tipo de nodo superior a su nodo existente.

5. Elija Save changes (Guardar cambios).

   El estado del clúster cambia a estado de modificación. Cuando el estado cambie a available (disponible), la modificación se habrá completado y podrá empezar a utilizar el nuevo clúster.

Ampliación de clústeres de MemoryDB (CLI) AWS

El siguiente procedimiento describe cómo escalar verticalmente un clúster de MemoryDB mediante la AWS CLI. Durante este proceso, el clúster de MemoryDB seguirá atendiendo solicitudes con un tiempo de inactividad mínimo.

Para escalar verticalmente a un clúster de MemoryDB (CLI) AWS

1. Determine los tipos de nodos a los que puede ampliar ejecutando el comando AWS CLI de la `list-allowed-node-type-updates` con el siguiente parámetro.

   Para Linux, macOS o Unix:

```
aws memorydb list-allowed-node-type-updates \
     --cluster-name my-cluster-name
```

   Para Windows:

```
aws memorydb list-allowed-node-type-updates ^
     --cluster-name my-cluster-name
```

   La salida del comando anterior es similar a la siguiente (formato JSON).

```
{
     "ScaleUpNodeTypes": [
         "db.r6g.2xlarge",
         "db.r6g.large"
     ],
     "ScaleDownNodeTypes": [
         "db.r6g.large"
     ],
```

```
}
```

Para obtener más información, consulte list-allowed-node-type-updates en la Referencia de AWS CLI.

2. Modifique su clúster para escalar verticalmente al nuevo tipo de nodo más grande con el comando de la AWS CLI `update-cluster` y los siguientes parámetros.

   - `--cluster-name`: nombre del clúster que está escalando verticalmente.
   - `--node-type`: tipo de nodo nuevo al que desea escalar el clúster. Este valor debe ser uno de los tipos de nodos devueltos por el comando `list-allowed-node-type-updates` en el paso 1.

   Para Linux, macOS o Unix:

   ```
   aws memorydb update-cluster  \
    --cluster-name my-cluster \
    --node-type db.r6g.2xlarge
   ```

   Para Windows:

   ```
   aws memorydb update-cluster ^
        --cluster-name my-cluster ^
        --node-type db.r6g.2xlarge ^
   ```

   Para obtener más información, consulte update-cluster.

Ampliación de clústeres de MemoryDB (API de MemoryDB)

El siguiente proceso escala su clúster de su tipo de nodo actual a un nuevo tipo de nodo más grande utilizando la API de MemoryDB. Durante este proceso, MemoryDB actualiza las entradas de DNS para que apunten a los nuevos nodos. Puede escalar clústeres con la conmutación por error habilitada mientras el clúster permanece en línea y atiende solicitudes de entrada.

El tiempo que se tarda en el escalado vertical a un tipo de nodo más grande varía en función de su tipo de nodo y de la cantidad de datos de su clúster actual.

Para escalar verticalmente a un clúster de MemoryDB (API de MemoryDB)

1. Determine qué tipos de nodos puede escalar verticalmente usando la acción `ListAllowedNodeTypeUpdates` de la API de MemoryDB con el siguiente parámetro.

   - `ClusterName`: el nombre del clúster. Use este parámetro para describir un clúster específico en lugar de todos los clústeres.

   ```
   https://memory-db.us-east-1.amazonaws.com/
       ?Action=ListAllowedNodeTypeUpdates
       &ClusterName=MyCluster
       &Version=2021-01-01
       &SignatureVersion=4
       &SignatureMethod=HmacSHA256
       &Timestamp=20210802T192317Z
       &X-Amz-Credential=<credential>
   ```

   Para obtener más información, consulte ListAllowedNodeTypeUpdates en la Referencia de la API de MemoryDB para Redis.

2. Escale verticalmente su clúster actual al nuevo tipo de nodo utilizando la acción `UpdateCluster` de la API de MemoryDB con los siguientes parámetros.

   - `ClusterName`: el nombre del clúster.
   - `NodeType`: el nuevo tipo de nodo más grande de clústeres en este clúster. Este valor debe ser uno de los tipos de instancia devueltos por la acción `ListAllowedNodeTypeUpdates` en el paso 1.

   ```
   https://memory-db.us-east-1.amazonaws.com/
       ?Action=UpdateCluster
       &NodeType=db.r6g.2xlarge
       &ClusterName=myCluster
       &SignatureVersion=4
       &SignatureMethod=HmacSHA256
       &Timestamp=20210801T220302Z
       &Version=2021-01-01
       &X-Amz-Algorithm=Amazon4-HMAC-SHA256
       &X-Amz-Date=20210801T220302Z
       &X-Amz-SignedHeaders=Host
       &X-Amz-Expires=20210801T220302Z
   ```

```
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Para obtener más información, consulte UpdateCluster.

Reducción vertical en línea

Temas

- Reducción vertical de clústeres de MemoryDB (consola)
- Reducir el tamaño de los clústeres de MemoryDB (CLI) AWS
- Reducir el tamaño de los clústeres de MemoryDB (API de MemoryDB)

Reducción vertical de clústeres de MemoryDB (consola)

El siguiente procedimiento describe cómo reducir verticalmente un clúster de MemoryDB de un único nodo mediante la AWS Management Console. Durante este proceso, el clúster de MemoryDB seguirá atendiendo solicitudes con un tiempo de inactividad mínimo.

Para reducir verticalmente un clúster de MemoryDB (consola)

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. En la lista de clústeres, elija el clúster preferido.
3. Elija Actions (Acciones) y después Modify (Modificar).
4. En el cuadro de diálogo Modificar clúster:
    - Elija el tipo de nodo que desee ampliar en la lista Node type. Para aplicar el escalado descendente, seleccione un tipo de nodo inferior a su nodo existente. Tenga en cuenta que no todos los tipos de nodos están disponibles para el proceso de reducción.
5. Elija Save changes (Guardar cambios).

    El estado del clúster cambia a estado de modificación. Cuando el estado cambie a available (disponible), la modificación se habrá completado y podrá empezar a utilizar el nuevo clúster.

Reducir el tamaño de los clústeres de MemoryDB (CLI) AWS

El siguiente procedimiento describe cómo reducir verticalmente un clúster de MemoryDB de un único nodo mediante la AWS CLI. Durante este proceso, el clúster de MemoryDB seguirá atendiendo solicitudes con un tiempo de inactividad mínimo.

Para reducir verticalmente a un clúster de MemoryDB (CLI) AWS

1. Determine los tipos de nodos a los que puede aplicar el escalado descendente ejecutando el comando AWS CLI de la `list-allowed-node-type-updates` con el siguiente parámetro.

   Para Linux, macOS o Unix:

   ```
   aws memorydb list-allowed-node-type-updates \
        --cluster-name my-cluster-name
   ```

   Para Windows:

   ```
   aws memorydb list-allowed-node-type-updates ^
        --cluster-name my-cluster-name
   ```

   La salida del comando anterior es similar a la siguiente (formato JSON).

   ```
   {
        "ScaleUpNodeTypes": [
            "db.r6g.2xlarge",
            "db.r6g.large"
        ],
        "ScaleDownNodeTypes": [
            "db.r6g.large"
        ],
   }
   ```

   Para obtener más información, consulte list-allowed-node-type-updates.

2. Modifique su clúster para reducirlo verticalmente al nuevo tipo de nodo más pequeño con el comando `update-cluster` y los siguientes parámetros.

   - `--cluster-name`: nombre del clúster que se reduce verticalmente.

- `--node-type`: tipo de nodo nuevo al que desea escalar el clúster. Este valor debe ser uno de los tipos de nodos devueltos por el comando `list-allowed-node-type-updates` en el paso 1.

Para Linux, macOS o Unix:

```
aws memorydb update-cluster  \
     --cluster-name my-cluster \
     --node-type db.r6g.large
```

Para Windows:

```
aws memorydb update-cluster ^
     --cluster-name my-cluster ^
     --node-type db.r6g.large
```

Para obtener más información, consulte update-cluster.

Reducir el tamaño de los clústeres de MemoryDB (API de MemoryDB)

El siguiente proceso escala su clúster de su tipo de nodo actual a un nuevo tipo de nodo más pequeño utilizando la API de MemoryDB. Durante este proceso, el clúster de MemoryDB seguirá atendiendo solicitudes con un tiempo de inactividad mínimo.

El tiempo que se tarda en la realización del escalado descendente a un tipo de nodo más pequeño varía en función de su tipo de nodo y de la cantidad de datos de su clúster actual.

Reducción vertical (API de MemoryDB)

1. Determine a qué tipos de nodos puede aplicar la reducción vertical usando la acción ListAllowedNodeTypeUpdates de la API con el siguiente parámetro:

   - `ClusterName`: el nombre del clúster. Use este parámetro para describir un clúster específico en lugar de todos los clústeres.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=ListAllowedNodeTypeUpdates
    &ClusterName=MyCluster
```

```
    &Version=2021-01-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210802T192317Z
    &X-Amz-Credential=<credential>
```

2. Aplique la reducción vertical a su clúster actual al nuevo tipo de nodo utilizando la acción UpdateCluster de la API y con los siguientes parámetros.

   - `ClusterName`: el nombre del clúster.
   - `NodeType`: el nuevo tipo de nodo más pequeño de clústeres en este clúster. Este valor debe ser uno de los tipos de instancia devueltos por la acción `ListAllowedNodeTypeUpdates` en el paso 1.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=UpdateCluster
    &NodeType=db.r6g.2xlarge
    &ClusterName=myReplGroup
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210801T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Configuración de los parámetros de motor mediante los grupos de parámetros

MemoryDB para Redis usa parámetros para controlar las propiedades de tiempo de ejecución de sus nodos y clústeres. Por lo general, las versiones de motor más reciente incluyen parámetros adicionales para ofrecer compatibilidad con la funcionalidad más reciente. Para ver las tablas de parámetros, consulte Parámetros específicos de Redis.

Como cabe esperar, determinados valores de parámetros, como `maxmemory`, dependen del tipo de nodo y de motor. Para ver una tabla de estos valores de los parámetros por tipo de nodo, consulte Parámetros específicos de tipo de nodo de MemoryDB.

Temas

Administración de parámetros

Los parámetros se agrupan en grupos de parámetros identificados para facilitar la administración de parámetros. Un grupo de parámetros representa una combinación de valores específicos de parámetros que se pasan al software del motor durante el startup. Estos valores determinan cómo se comportan los procesos del motor en cada nodo durante el tiempo de ejecución. Los valores de parámetros de un grupo de parámetros determinado se aplican a todos los nodos asociados al grupo, independientemente del clúster al que pertenezcan.

Para ajustar el rendimiento del clúster, puede modificar los valores de algunos parámetros o cambiar el grupo de parámetros del clúster.

- No puede modificar ni eliminar los grupos de parámetros predeterminados. Si necesita valores de parámetros personalizados, debe crear un grupo de parámetros personalizado.
- La familia del grupo parámetros y el clúster que va a asignar deben ser compatibles. Por ejemplo, si el clúster ejecuta la versión 6 de Redis, solo se pueden utilizar los grupos de parámetros, predeterminados o personalizados, de la familia memorydb_redis6.
- Cuando cambia los parámetros de un clúster, el cambio se aplica al clúster inmediatamente. Esto es cierto tanto si se modifica el propio grupo de parámetros del clúster como si se modifica el valor de un parámetro del grupo.

Niveles de grupo de parámetros

Niveles de grupos de parámetros de MemoryDB para Redis

Predeterminado global

Este es el grupo de parámetros raíz de nivel superior para todos los clientes de MemoryDB para Redis de la región.

Grupo de parámetros predeterminado global:

- Está reservado para MemoryDB y no está disponible para el cliente.

Predeterminado del cliente

Se trata de una copia del grupo de parámetros predeterminado global que se crea para el uso del cliente.

Grupo de parámetros predeterminado del cliente:

- Se crea mediante MemoryDB y es de su propiedad.
- Está disponible para el cliente para el uso como grupo de parámetros para cualquier clúster que ejecute una versión del motor compatible con este grupo de parámetros.
- No admite modificación del cliente.

Propiedad del cliente

Se trata de una copia del grupo de parámetros predeterminado del cliente. Se crea un grupo de parámetros Propiedad del cliente cuando el cliente crea un grupo de parámetros.

Grupo de parámetros propiedad del cliente:

- Lo crea el cliente y es de su propiedad.
- Puede asignarse a cualquiera de los clústeres compatibles del cliente.
- El cliente puede modificarlo para crear un nuevo grupo de parámetros personalizado

No todos los valores de parámetros se pueden modificar. Para obtener más información, consulte Parámetros específicos de Redis.

Creación de un grupo de parámetros

Debe crear un nuevo grupo de parámetros si existe uno o varios parámetros que desee cambiar con respecto a los valores predeterminados. Puede crear un grupo de parámetros mediante la consola MemoryDB AWS CLI, la o la API MemoryDB.

Creación de un grupo de parámetros (consola)

En el siguiente procedimiento se muestra cómo crear un grupo de parámetros mediante la consola de MemoryDB.

Para crear un grupo de parámetros con la consola de MemoryDB

1. Inicie sesión en la consola MemoryDB AWS Management Console for Redis y ábrala en https://console.aws.amazon.com/memorydb/.
2. Para ver una lista de todos los grupos de parámetros disponibles, en el panel de navegación izquierdo, elija Parameter Groups.
3. Para crear un grupo de parámetros, elija Crear grupo de parámetros.

   Aparece la página Crear grupo de parámetros.
4. En el cuadro Name, escriba un nombre único para este grupo de parámetros.

   Al crear un clúster o modificar un grupo de parámetros de clúster, podrá elegir el grupo de parámetros por su nombre. Por lo tanto, se recomienda que el nombre sea informativo y que identifique de algún modo la familia del grupo de parámetros.

   Las restricciones de nomenclatura de los grupos de parámetros son las siguientes:

   - Deben comenzar por una letra ASCII.
   - Solo puede contener letras ASCII, dígitos y guiones.
   - Debe tener de 1 a 255 caracteres.
   - No pueden contener dos guiones consecutivos.
   - No pueden terminar con un guion.
5. En el cuadro Description, escriba una descripción para el grupo de parámetros.
6. En el cuadro Compatibilidad de versiones de Redis, elija una versión del motor a la que corresponda este grupo de parámetros.

7. En las etiquetas, si lo desea, añada etiquetas para buscar y filtrar sus grupos de parámetros o realizar un seguimiento de sus costes. AWS
8. Para crear el grupo de parámetros, elija Create.

   Para finalizar el proceso sin crear el grupo de parámetros, seleccione Cancel.
9. Cuando se cree el grupo de parámetros, tendrá los valores predeterminados de la familia. Para cambiar los valores predeterminados, debe modificar el grupo de parámetros. Para obtener más información, consulte Modificación de un grupo de parámetros.

Creación de un grupo de parámetros (AWS CLI)

Para crear un grupo de parámetros mediante el AWS CLI, utilice el comando `create-parameter-group` con estos parámetros.

- `--parameter-group-name`: el nombre del grupo de parámetros.

  Las restricciones de nomenclatura de los grupos de parámetros son las siguientes:
  - Deben comenzar por una letra ASCII.
  - Solo puede contener letras ASCII, dígitos y guiones.
  - Debe tener de 1 a 255 caracteres.
  - No pueden contener dos guiones consecutivos.
  - No pueden terminar con un guion.
- `--family`: la familia de versión y motor del grupo de parámetros.
- `--description`: una descripción del usuario para el grupo de parámetros.

Example

En el ejemplo siguiente se crea un grupo de parámetros denominado myRedis6x que usa la familia memorydb_redis6 como plantilla.

Para Linux, macOS o Unix:

```
aws memorydb create-parameter-group \
    --parameter-group-name myRedis6x  \
    --family memorydb_redis6 \
    --description "My first parameter group"
```

Para Windows:

```
aws memorydb create-parameter-group ^
    --parameter-group-name myRedis6x  ^
    --family memorydb_redis6 ^
    --description "My first parameter group"
```

La salida de este comando será similar a lo que se muestra a continuación.

```
{
    "ParameterGroup": {
        "Name": "myRedis6x",
        "Family": "memorydb_redis6",
        "Description": "My first parameter group",
        "ARN": "arn:aws:memorydb:us-east-1:012345678912:parametergroup/myredis6x"
    }
}
```

Cuando se cree el grupo de parámetros, tendrá los valores predeterminados de la familia. Para cambiar los valores predeterminados, debe modificar el grupo de parámetros. Para obtener más información, consulte Modificación de un grupo de parámetros.

Para obtener más información, consulte `create-parameter-group`.

Creación de un grupo de parámetros (API de MemoryDB)

Para crear un grupo de parámetros con la API de MemoryDB, use la acción `CreateParameterGroup` con los parámetros que se indican a continuación.

- `ParameterGroupName`: el nombre del grupo de parámetros.

  Las restricciones de nomenclatura de los grupos de parámetros son las siguientes:

  - Deben comenzar por una letra ASCII.
  - Solo puede contener letras ASCII, dígitos y guiones.
  - Debe tener de 1 a 255 caracteres.
  - No pueden contener dos guiones consecutivos.
  - No pueden terminar con un guion.
- `Family`: la familia de versión y motor del grupo de parámetros. Por ejemplo, `memorydb_redis6`.
- `Description`: una descripción del usuario para el grupo de parámetros.

Example

En el ejemplo siguiente se crea un grupo de parámetros denominado myRedis6x que usa la familia memorydb_redis6 como plantilla.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=CreateParameterGroup
   &Family=memorydb_redis6
   &ParameterGroupName=myRedis6x
   &Description=My%20first%20parameter%20group
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20210802T192317Z
   &Version=2021-01-01
   &X-Amz-Credential=<credential>
```

La respuesta a esta acción será similar a lo que se muestra a continuación.

```
<CreateParameterGroupResponse xmlns="http://memory-db.us-east-1.amazonaws.com/
doc/2021-01-01/">
  <CreateParameterGroupResult>
    <ParameterGroup>
      <Name>myRedis6x</Name>
      <Family>memorydb_redis6</Family>
      <Description>My first parameter group</Description>
      <ARN>arn:aws:memorydb:us-east-1:012345678912:parametergroup/myredis6x</ARN>
    </ParameterGroup>
  </CreateParameterGroupResult>
  <ResponseMetadata>
    <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>
  </ResponseMetadata>
</CreateParameterGroupResponse>
```

Cuando se cree el grupo de parámetros, tendrá los valores predeterminados de la familia. Para cambiar los valores predeterminados, debe modificar el grupo de parámetros. Para obtener más información, consulte Modificación de un grupo de parámetros.

Para obtener más información, consulte `CreateParameterGroup`.

Enumeración de grupos de parámetros por nombre

Puede enumerar los grupos de parámetros mediante la consola MemoryDB AWS CLI, la o la API MemoryDB.

Enumeración de grupos de parámetros por nombre (consola)

En el siguiente procedimiento se muestra cómo ver una lista de grupos de parámetros mediante la consola de MemoryDB.

Para obtener una lista con los grupos de parámetros mediante la consola de MemoryDB

1. Inicie sesión en la consola MemoryDB AWS Management Console for Redis y ábrala en https://console.aws.amazon.com/memorydb/.
2. Para ver una lista de todos los grupos de parámetros disponibles, en el panel de navegación izquierdo, elija Parameter Groups.

Listado de grupos de parámetros por nombre (AWS CLI)

Para generar una lista de grupos de parámetros mediante el AWS CLI, utilice el comando`describe-parameter-groups`. Si proporciona un nombre de grupo de parámetros, solo se mostrará el grupo de parámetros de dicho nombre. Si no proporciona ningún nombre de grupo de parámetros, se mostrarán hasta `--max-results` grupos de parámetros. En cualquier caso, se mostrarán el nombre, la familia y la descripción del grupo de parámetros.

Example

El siguiente código de ejemplo muestra el grupo de parámetros myRedis6x.

Para Linux, macOS o Unix:

```
aws memorydb describe-parameter-groups \
    --parameter-group-name myRedis6x
```

Para Windows:

```
aws memorydb describe-parameter-groups ^
    --parameter-group-name myRedis6x
```

La salida de este comando tendrá un aspecto similar al siguiente y mostrará el nombre, la familia y la descripción del grupo de parámetros.

```
{
    "ParameterGroups": [
     {
         "Name": "myRedis6x",
         "Family": "memorydb_redis6",
         "Description": "My first parameter group",
                "ARN": "arn:aws:memorydb:us-east-1:012345678912:parametergroup/
myredis6x"
     }
    ]
}
```

Example

El siguiente código de muestra indica el grupo de parámetros myRedis6x para grupos de parámetros que se ejecutan en la versión 5.0.6 y siguientes del motor de Redis.

Para Linux, macOS o Unix:

```
aws memorydb describe-parameter-groups \
    --parameter-group-name myRedis6x
```

Para Windows:

```
aws memorydb describe-parameter-groups ^
    --parameter-group-name myRedis6x
```

La salida de este comando tendrá un aspecto similar al siguiente y mostrará el nombre, la familia y la descripción del grupo de parámetros.

```
{
    "ParameterGroups": [
     {
         "Name": "myRedis6x",
         "Family": "memorydb_redis6",
         "Description": "My first parameter group",
                "ARN": "arn:aws:memorydb:us-east-1:012345678912:parametergroup/
myredis6x"
```

```
     }
    ]
}
```

Example

El siguiente código de ejemplo muestra hasta 20 grupos de parámetros.

```
aws memorydb describe-parameter-groups --max-results 20
```

La salida JSON de este comando tendrá un aspecto similar al siguiente y mostrará el nombre, la familia y la descripción de cada grupo de parámetros.

```
{
    "ParameterGroups": [
        {
            "ParameterGroupName": "default.memorydb-redis6",
            "Family": "memorydb_redis6",
            "Description": "Default parameter group for memorydb_redis6",
            "ARN": "arn:aws:memorydb:us-east-1:012345678912:parametergroup/
default.memorydb-redis6"
        },
        ...
    ]
}
```

Para obtener más información, consulte [describe-parameter-groups](#).

Lista de grupos de parámetros por nombre (API de MemoryDB)

Para generar una lista de grupos de parámetros mediante la API de MemoryDB, use la acción `DescribeParameterGroups`. Si proporciona un nombre de grupo de parámetros, solo se mostrará el grupo de parámetros de dicho nombre. Si no proporciona ningún nombre de grupo de parámetros, se mostrarán hasta `MaxResults` grupos de parámetros. En cualquier caso, se mostrarán el nombre, la familia y la descripción del grupo de parámetros.

Example

El siguiente código de ejemplo muestra hasta 20 grupos de parámetros.

```
https://memory-db.us-east-1.amazonaws.com/
```

```
   ?Action=DescribeParameterGroups
   &MaxResults=20
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20210802T192317Z
   &Version=2021-01-01
   &X-Amz-Credential=<credential>
```

La respuesta de esta acción tendrá un aspecto similar al siguiente y mostrará el nombre, la familia y la descripción de cada grupo de parámetros en el caso de memorydb_redis6.

```
<DescribeParameterGroupsResponse xmlns="http://memory-db.us-east-1.amazonaws.com/
doc/2021-01-01/">
  <DescribeParameterGroupsResult>
    <ParameterGroups>
      <ParameterGroup>
        <Name>myRedis6x</Name>
        <Family>memorydb_redis6</Family>
        <Description>My custom Redis 6 parameter group</Description>
        <ARN>arn:aws:memorydb:us-east-1:012345678912:parametergroup/myredis6x</ARN>
      </ParameterGroup>
       <ParameterGroup>
        <Name>default.memorydb-redis6</Name>
        <Family>memorydb_redis6</Family>
        <Description>Default parameter group for memorydb_redis6</Description>
        <ARN>arn:aws:memorydb:us-east-1:012345678912:parametergroup/default.memorydb-
redis6</ARN>
      </ParameterGroup>
    </ParameterGroups>
  </DescribeParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeParameterGroupsResponse>
```

Example

El siguiente código de ejemplo muestra el grupo de parámetros myRedis6x.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DescribeParameterGroups
   &ParameterGroupName=myRedis6x
   &SignatureVersion=4
```

```
   &SignatureMethod=HmacSHA256
   &Timestamp=20210802T192317Z
   &Version=2021-01-01
   &X-Amz-Credential=<credential>
```

La respuesta de esta acción tendrá un aspecto similar al siguiente y mostrará el nombre, la familia y la descripción.

```
<DescribeParameterGroupsResponse xmlns="http://memory-db.us-east-1.amazonaws.com/
doc/2021-01-01/">
  <DescribeParameterGroupsResult>
    <ParameterGroups>
      <ParameterGroup>
        <Name>myRedis6x</Name>
        <Family>memorydb_redis6</Family>
        <Description>My custom Redis 6 parameter group</Description>
        <ARN>arn:aws:memorydb:us-east-1:012345678912:parametergroup/myredis6x</ARN>
      </ParameterGroup>
    </ParameterGroups>
  </DescribeParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeParameterGroupsResponse>
```

Para obtener más información, consulte DescribeParameterGroups.

Enumeración de valores de un grupo de parámetros

Puede enumerar los parámetros y sus valores para un grupo de parámetros mediante la consola MemoryDB AWS CLI, la o la API MemoryDB.

Enumeración de valores de un grupo de parámetros (consola)

El procedimiento siguiente muestra cómo obtener una lista de los parámetros de un grupo de parámetros, junto con sus valores, mediante la consola de MemoryDB.

Para obtener una lista de los parámetros de un grupo de parámetros, junto con sus valores, mediante la consola de MemoryDB

1. Inicie sesión en la consola MemoryDB AWS Management Console for Redis y ábrala en https://console.aws.amazon.com/memorydb/.
2. Para ver una lista de todos los grupos de parámetros disponibles, en el panel de navegación izquierdo, elija Parameter Groups.
3. Elija el grupo de parámetros del que desea obtener una lista de los parámetros y sus valores eligiendo el nombre (no la casilla situada a su lado) del nombre del grupo de parámetros.

   Los parámetros y sus valores se mostrarán en la parte inferior de la pantalla. Debido al número de parámetros, puede que tenga que desplazarse hacia arriba y hacia abajo para encontrar el parámetro que le interesa.

Listar los valores de un grupo de parámetros (AWS CLI)

Para enumerar los parámetros de un grupo de parámetros y sus valores mediante el AWS CLI, utilice el comando`describe-parameters`.

Example

El siguiente código de ejemplo muestra todos los parámetros, junto con sus valores, del grupo de parámetros myRedis6x.

Para Linux, macOS o Unix:

```
aws memorydb describe-parameters \
    --parameter-group-name myRedis6x
```

Para Windows:

```
aws memorydb describe-parameters ^
    --parameter-group-name myRedis6x
```

Para obtener más información, consulte describe-parameters.

Lista de valores de un grupo de parámetros (API de MemoryDB)

Para obtener una lista de los parámetros de un grupo de parámetros, junto con sus valores, mediante la API de MemoryDB, use la acción `DescribeParameters`.

Para obtener más información, consulte DescribeParameters.

Modificación de un grupo de parámetros

> **Important**
>
> No es posible modificar ningún grupo de parámetros predeterminado.

Es posible modificar algunos parámetros de un grupo de parámetros. Dichos valores de parámetros se aplican a los clústeres asociados al grupo de parámetros. Para obtener más información acerca de cuándo se aplica un cambio en los valores de los parámetros a un grupo de parámetros, consulte Parámetros específicos de Redis.

Modificación de un grupo de parámetros (consola)

En el siguiente procedimiento se muestra cómo cambiar el valor del parámetro mediante la consola de MemoryDB. Puede usar el mismo procedimiento para cambiar el valor de cualquier parámetro.

Para cambiar el valor de un parámetro mediante la consola de MemoryDB

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. Para ver una lista de todos los grupos de parámetros disponibles, en el panel de navegación izquierdo, elija Parameter Groups.
3. Seleccione el grupo de parámetros que desea modificar eligiendo el botón de opción situado a la izquierda del nombre del grupo de parámetros.

Elija Acciones y, a continuación, Ver detalles. Como alternativa, también puede elegir el nombre del grupo de parámetros para ir a la página de detalles.

4. Para modificar el parámetro, elija Editar. Se habilitará la edición de todos los parámetros editables. Puede que tenga que desplazarse por las páginas para encontrar el parámetro que desea cambiar. También puede buscar el parámetro por nombre, valor o tipo en el cuadro de búsqueda.
5. Realice las modificaciones necesarias en los parámetros.
6. Para guardar los cambios, elija Guardar cambios.
7. Si ha modificado los valores de los parámetros a lo largo del número de páginas, puede revisar todos los cambios seleccionando Vista previa de los cambios. Para confirmar los cambios, elija Guardar cambios. Para realizar más modificaciones, seleccione Atrás.
8. La página Detalles de los parámetros también ofrece la opción de restablecer los valores predeterminados. Para restablecer los valores predeterminados, seleccione Restablecer los valores predeterminados. Las casillas de verificación se muestran en el lado izquierdo de todos los parámetros. Puede seleccionar los que desee restablecer y elegir Continuar con el restablecimiento para confirmarlos.

   Elija confirmar para confirmar la acción de restablecimiento en el cuadro de diálogo.
9. La página de detalles de los parámetros le permite establecer el número de parámetros que desea ver en cada página. Use la rueda dentada del lado derecho para realizar esos cambios. También puede activar o desactivar las columnas que desee en la página de detalles. Estos cambios perduran durante toda la sesión de la consola.

   Para encontrar el nombre del parámetro que ha cambiado, consulte Parámetros específicos de Redis.

Modificación de un grupo de parámetros (AWS CLI)

Para cambiar el valor de un parámetro mediante el AWS CLI, utilice el comando`update-parameter-group`.

Para encontrar el nombre y los valores permitidos del parámetro que desea cambiar, consulte Parámetros específicos de Redis.

Para obtener más información, consulte update-parameter-group.

Modificación de un grupo de parámetros (API de MemoryDB)

Para cambiar los valores de un grupo de parámetros usando la API de MemoryDB, use la acción `UpdateParameterGroup`.

Para encontrar el nombre y los valores permitidos del parámetro que desea cambiar, consulte Parámetros específicos de Redis.

Para obtener más información, consulte `UpdateParameterGroup`.

Eliminación de un grupo de parámetros

Puede eliminar un grupo de parámetros personalizado mediante la consola de MemoryDB AWS CLI, la o la API de MemoryDB.

No podrá eliminar un grupo de parámetros si está asociado a un clúster. Tampoco podrá eliminar ninguno de los grupos de parámetros predeterminados.

Eliminación de un grupo de parámetros (consola)

En el siguiente procedimiento se muestra cómo eliminar un grupo de parámetros mediante la consola de MemoryDB.

Para eliminar un grupo de parámetros con la consola de MemoryDB

1. Inicie sesión en la consola MemoryDB AWS Management Console for Redis y ábrala en https://console.aws.amazon.com/memorydb/.
2. Para ver una lista de todos los grupos de parámetros disponibles, en el panel de navegación izquierdo, elija Parameter Groups.
3. Seleccione el grupo de parámetros que desea eliminar eligiendo el botón de opción situado a la izquierda del nombre del grupo de parámetros.

   Elija Acciones y, a continuación, elija Eliminar.
4. Aparecerá la pantalla de confirmación Delete Parameter Groups.
5. Para eliminar los grupos de parámetros, introduzca Eliminar en el cuadro de texto de confirmación.

   Para conservar los grupos de parámetros, elija Cancel.

Eliminar un grupo de parámetros (AWS CLI)

Para eliminar un grupo de parámetros mediante el AWS CLI, utilice el comando`delete-parameter-group`. Para que el grupo de parámetros se elimine, el grupo de parámetros especificado mediante `--parameter-group-name` no puede tener ningún clúster asociado al grupo ni puede ser un grupo de parámetros predeterminado.

El siguiente código de muestra elimina el grupo de parámetros myRedis6x.

Example

Para Linux, macOS o Unix:

```
aws memorydb delete-parameter-group \
    --parameter-group-name myRedis6x
```

Para Windows:

```
aws memorydb delete-parameter-group ^
    --parameter-group-name myRedis6x
```

Para obtener más información, consulte delete-parameter-group.

Eliminación de un grupo de parámetros (API de MemoryDB)

Para eliminar un grupo de parámetros mediante la API de MemoryDB, use la acción `DeleteParameterGroup`. Para que el grupo de parámetros se elimine, el grupo de parámetros especificado mediante `ParameterGroupName` no puede tener ningún clúster asociado al grupo ni puede ser un grupo de parámetros predeterminado.

Example

El siguiente código de muestra elimina el grupo de parámetros myRedis6x.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DeleteParameterGroup
   &ParameterGroupName=myRedis6x
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &Timestamp=20210802T192317Z
   &Version=2021-01-01
   &X-Amz-Credential=<credential>
```

Para obtener más información, consulte `DeleteParameterGroup`.

Parámetros específicos de Redis

Si no se especifica ningún grupo de parámetros para el clúster de Redis, se usará un grupo de parámetros predeterminado apropiado para la versión del motor. No puede cambiar los valores de los parámetros de un grupo de parámetros predeterminado. Sin embargo, puede crear un grupo de parámetros personalizado y asignarlo a su clúster en cualquier momento, siempre y cuando los valores de los parámetros modificables condicionalmente sean iguales en ambos grupos de parámetros. Para obtener más información, consulte Creación de un grupo de parámetros.

Temas

Cambios de parámetros de Redis 7

> **Note**
>
> MemoryDB ha presentado una versión preliminar de la búsqueda vectorial que incluye un nuevo grupo de parámetros inmutables `default.memorydb-redis7.search.preview`. Este grupo de parámetros está disponible en la consola de MemoryDB y al crear un vector-search-enabled clúster nuevo mediante el comando CLI create-cluster. La versión preliminar está disponible en las siguientes AWS regiones: EE.UU. Este (Norte de Virginia), EE.UU. Este (Ohio), EE.UU. Oeste (Oregón), Asia Pacífico (Tokio) y Europa (Irlanda).

Familia de grupos de parámetros: memorydb_redis7

Los parámetros agregados en Redis 7 son los siguientes.

Nombre	Detalles	Descripción
`latency-tracking`	Valores permitidos: `yes`, `no` Valor predeterminado: `no` Tipo: cadena	Cuando se establece en sí, realiza un seguimiento de las latencias por comando y permite exportar la distribución de percentiles mediante el comando de estadísticas de latencia `INFO` y las distribuciones de latencia

Nombre	Detalles	Descripción
	Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	acumulada (histogramas) mediante el comando `LATENCY`.
`hash-max-listpack-entries`	Valores permitidos: `0+` Valor predeterminado: `512` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	El número máximo de entradas de hash para comprimir el conjunto de datos.
`hash-max-listpack-value`	Valores permitidos: `0+` Valor predeterminado: `64` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	El umbral de entradas de hash más grandes para comprimir el conjunto de datos.

Nombre	Detalles	Descripción
`zset-max-listpack-entries`	Valores permitidos: `0+` Valor predeterminado: `128` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	El número máximo de entradas de conjuntos ordenados para comprimir el conjunto de datos.
`zset-max-listpack-value`	Valores permitidos: `0+` Valor predeterminado: `64` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	El umbral de entradas de conjuntos ordenados más grandes para comprimir el conjunto de datos.

Los parámetros cambiados en Redis 7 son los siguientes.

Nombre	Detalles	Descripción
`activerehashing`	Modificable: no. En Redis 7, este parámetro está oculto y habilitado de forma predeterminada. Para desactivarlo, debe crear un caso de soporte.	Modificable era sí.

Los parámetros eliminados de Redis 7 son los siguientes.

Nombre	Detalles	Descripción
`hash-max-ziplist-entries`	Valores permitidos: `0+` Valor predeterminado: `512` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	Use `listpack` en lugar de `ziplist` para representar una codificación de hash pequeña
`hash-max-ziplist-value`	Valores permitidos: `0+` Valor predeterminado: `64` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	Use `listpack` en lugar de `ziplist` para representar una codificación de hash pequeña
`zset-max-ziplist-entries`	Valores permitidos: `0+` Valor predeterminado: `128` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	Use `listpack` en lugar de `ziplist` para representar una codificación de hash pequeña.

Nombre	Detalles	Descripción
`zset-max-ziplist-value`	Valores permitidos: `0+` Valor predeterminado: `64` Tipo: número entero Modificable: sí Los cambios surten efecto: inmediatamente en todos los nodos del clúster.	Use `listpack` en lugar de `ziplist` para representar una codificación de hash pequeña.

Parámetros de Redis 6

> **Note**
>
> En la versión 6.2 del motor de Redis, cuando se introdujo la familia de nodos r6gd para su uso con Organización de datos en niveles, solo las políticas max-memory `noeviction`, `volatile-lru` y `allkeys-lru` se admiten con tipos de nodos r6gd.

Familia de grupos de parámetros: memorydb_redis6

Los parámetros agregados en Redis 6 son los siguientes.

Nombre	Detalles	Descripción
`maxmemory-policy`	Tipo: STRING Valores permitidos: volatile-lru, allkeys-lru, volatile-lfu, allkeys-lfu, volatile-random, allkeys-random, volatile-ttl, noeviction	Política de expulsión de claves cuando se alcanza el uso máximo de la memoria. Para obtener más información, consulte Uso de Redis como una caché de LRU.

Nombre	Detalles	Descripción
	El valor predeterminado es noeviction	
`list-compress-depth`	Tipo: INTEGER Valores permitidos: 0- Predeterminado: 0	La profundidad de compresión es el número de nodos de listas comprimidas de listas rápidas de ambos lados de la lista que se excluirán de la compresión. El principio y el final de la lista están siempre sin comprimir para agilizar las operaciones de inserción y extracción. Los valores son los siguientes: • 0: deshabilitar toda la compresión. • 1: comenzar a comprimir con el primer nodo a partir del principio y el final. [principio]->nodo->nodo->...->nodo->[final] Se comprimen todos los nodos excepto los nodos [principio] y [final]. • 2: comenzar a comprimir con el segundo nodo a partir del principio y el final. [principio]->[siguiente]->nodo->nodo->...->nodo->[penúltimo]->[final] Los nodos [principio], [siguiente], [penúltimo] y [final] no se comprimen. Todos los demás nodos se comprimen. • etc.

Nombre	Detalles	Descripción
`hll-sparse-max-bytes`	Tipo: INTEGER Valores permitidos: 1-16000 Predeterminado: 3000	HyperLogLog límite de bytes de representación dispersa. El límite incluye el encabezado de 16 bytes. Cuando el HyperLogLog uso de la representación dispersa cruza este límite, se convierte en la representación densa. No se recomienda usar un valor superior a 16 000, ya que en ese punto, la representación densa es más eficaz desde el punto de vista de la memoria. Se recomienda usar un valor próximo a 3000 con el fin de disponer de los beneficios de la codificación eficaz desde el punto de vista del espacio sin ralentizar demasiado `PFADD`, que está habilitado con la codificación dispersa. El valor se puede aumentar a ~10000 cuando la CPU no es un problema, pero sí el espacio, y el conjunto de datos está compuesto por muchos HyperLogLogs con una cardinalidad en el rango de 0 a 15000.
`lfu-log-factor`	Tipo: INTEGER Valores permitidos: 1- Predeterminado: 10	El factor logarítmico para incrementar el contador de claves para la política de desalojos de la LFU.
`lfu-decay-time`	Tipo: INTEGER Valores permitidos: 0- Valor predeterminado: 1	Tiempo en minutos para disminuir el contador de claves para la política de expulsión de LFU.

Nombre	Detalles	Descripción
`active-defrag-max-scan-fields`	Tipo: INTEGER Valores permitidos: 1-1000000 Predeterminado: 1000	Número máximo de campos set/hash/zset/list que se procesarán desde el análisis del diccionario principal durante la desfragmentación activa.
`active-defrag-threshold-upper`	Tipo: INTEGER Valores permitidos: 1-100 Predeterminado: 100	Porcentaje máximo de fragmentación en el que usará el máximo esfuerzo.
`client-output-buffer-limit-pubsub-hard-limit`	Tipo: INTEGER Valores permitidos: 0- Predeterminado: 33554432	Para clientes de publicación/suscripción de Redis: si el búfer de salida de un cliente alcanza el número de bytes especificado, el cliente se desconectará.
`client-output-buffer-limit-pubsub-soft-limit`	Tipo: INTEGER Valores permitidos: 0- Predeterminado: 8388608	Para clientes de publicación/suscripción de Redis: si el búfer de salida de un cliente alcanza el número de bytes especificado, el cliente se desconectará solo si esta condición se mantiene durante `client-output-buffer-limit-pubsub-soft-seconds.`
`client-output-buffer-limit-pubsub-soft-seconds`	Tipo: INTEGER Valores permitidos: 0- Predeterminado: 60	Para clientes de publicación/suscripción de Redis: si el búfer de salida de un cliente permanece en `client-output-buffer-limit-pubsub-soft-limit` bytes por un tiempo superior a este número de segundos, el cliente se desconectará.

Nombre	Detalles	Descripción
`timeout`	Tipo: INTEGER Valores permitidos: 0,20- Predeterminado: 0	Número de segundos que un nodo espera antes de caducar. Valores son los siguientes: • 0: no desconectar nunca un cliente inactivo. • 1-19: valores no válidos. • >=20: número de segundos que un nodo espera antes de desconectar un cliente inactivo.
`notify-keyspace-events`	Tipo: STRING Valores permitidos: NULO Valor predeterminado: NULO	Los eventos del espacio de claves sobre los que Redis debe notificar a los clientes de Pub/Sub. Todas las notificaciones están desactivadas de forma predeterminada.
`maxmemory-samples`	Tipo: INTEGER Valores permitidos: 1- Valor predeterminado: 3	Para `time-to-live (TTL)` los cálculos least-recently-used `(LRU)` y cálculos, este parámetro representa el tamaño de la muestra de las claves que se van a comprobar. De forma predeterminada, Redis elige 3 claves y usa la que se usó menos recientemente.
`slowlog-max-len`	Tipo: INTEGER Valores permitidos: 0- Valor predeterminado: 128	Tamaño máximo de la característica Slow Log de Redis. Esta longitud no tiene límite. Solo tenga en cuenta que consumirá memoria. Puede recuperar la memoria utilizada por el registro lento con `SLOWLOG RESET.`

Nombre	Detalles	Descripción
activerehashing	Tipo: STRING Valores permitidos: sí, no Valor predeterminado: yes	La tabla de hash principal se recombina diez veces por segundo; cada operación de recombinación consume 1 milisegundo de tiempo de procesamiento de la CPU. Este valor se establece al crear el grupo de parámetros. Cuando se asigne un nuevo grupo de parámetros a un clúster, este valor debe ser el mismo tanto en el nuevo grupo de parámetros como en el anterior.
client-output-buffer-limit-normal-hard-limit	Tipo: INTEGER Valores permitidos: 0- Predeterminado: 0	Si el búfer de salida de un cliente alcanza el número de bytes especificado, el cliente se desconectará. El valor predeterminado es cero (sin límite flexible).
client-output-buffer-limit-normal-soft-limit	Tipo: INTEGER Valores permitidos: 0- Predeterminado: 0	Si el búfer de salida de un cliente alcanza el número de bytes especificado, el cliente se desconectará solo si esta condición se mantiene durante client-output-buffer-limit-normal-soft-seconds . El valor predeterminado es cero (sin límite duro).
client-output-buffer-limit-normal-soft-seconds	Tipo: INTEGER Valores permitidos: 0- Predeterminado: 0	Si el búfer de salida de un cliente permanece en client-output-buffer-limit-normal-soft-limit bytes por un periodo superior a este número de segundos, el cliente se desconectará. El valor predeterminado es cero (sin límite de tiempo).

Nombre	Detalles	Descripción
`tcp-keepalive`	Tipo: INTEGER Valores permitidos: 0- Predeterminado: 300	Si se establecen un valor distinto de cero (N), los clientes de nodo se sondearán cada N segundos para asegurarse de que siguen conectados. Con el valor predeterminado 0, el sondeo se desactiva.
`active-defrag-cycle-min`	Tipo: INTEGER Valores permitidos: 1-75 Predeterminado: 5	Esfuerzo mínimo para desfragmentar en porcentaje de CPU.
`stream-node-max-bytes`	Tipo: INTEGER Valores permitidos: 0- Predeterminado: 4096	La estructura de datos de secuencia es un árbol de prefijos de nodos que contiene varios elementos. Utilice esta configuración para especificar el tamaño máximo de un único nodo de un árbol de prefijos in bytes. Si se establece en 0, el tamaño del nodo del árbol es ilimitado.
`stream-node-max-entries`	Tipo: INTEGER Valores permitidos: 0- Predeterminado: 100	La estructura de datos de secuencia es un árbol de prefijos de nodos que contiene varios elementos. Utilice esta configuración para especificar el número máximo de elementos que puede contener un único nodo antes de cambiar a un nodo nuevo al agregar entradas nuevas de secuencia. Si se establece en 0, el número de elementos del nodo del árbol es ilimitado.

Nombre	Detalles	Descripción
`lazyfree-lazy-eviction`	Tipo: STRING Valores permitidos: sí, no Valor predeterminado: no	Realiza una eliminación asíncrona en las expulsiones.
`active-defrag-ignore-bytes`	Tipo: INTEGER Valores permitidos: 1048576- Predeterminado: 104857600	Cantidad mínima de restos de fragmentación para comenzar la desfragmentación activa.
`lazyfree-lazy-expire`	Tipo: STRING Valores permitidos: sí, no Valor predeterminado: no	Realiza una eliminación asíncrona en las claves vencidas.
`active-defrag-threshold-lower`	Tipo: INTEGER Valores permitidos: 1-100 Predeterminado: 10	Porcentaje mínimo de fragmentación para comenzar la desfragmentación activa.
`active-defrag-cycle-max`	Tipo: INTEGER Valores permitidos: 1-75 Predeterminado: 75	Esfuerzo máximo para desfragmentar en porcentaje de CPU.
`lazyfree-lazy-server-del`	Tipo: STRING Valores permitidos: sí, no Valor predeterminado: no	Realiza una eliminación asíncrona de los comandos que actualizan valores.

Nombre	Detalles	Descripción
`slowlog-log-slower-than`	Tipo: INTEGER Valores permitidos: 0- Predeterminado: 10000	Tiempo de ejecución máximo, en microsegundos, que debe superarse para que los comandos se registren con la característica `Slow Log` de Redis. Tenga en cuenta que un número negativo desactiva el registro lento, mientras que un valor de cero fuerza el registro de todos los comandos.
`hash-max-ziplist-entries`	Tipo: INTEGER Valores permitidos: 0- Predeterminado: 512	Determina la cantidad de memoria que usan los hash. Los hash con un número de entradas inferior al especificado se almacenan con una codificación especial que permite ahorrar espacio.
`hash-max-ziplist-value`	Tipo: INTEGER Valores permitidos: 0- Predeterminado: 64	Determina la cantidad de memoria que usan los hash. Los hash con entradas de tamaño inferior al número de bytes especificado se almacenan con una codificación especial que permite ahorrar espacio.
`set-max-intset-entries`	Tipo: INTEGER Valores permitidos: 0- Predeterminado: 512	Determina la cantidad de memoria que se usa para determinados tipos de conjuntos (cadenas que son enteros en base 10 en el rango de enteros con signo de 64 bits). Estos conjuntos con un número de entradas inferior al especificado se almacenan con una codificación especial que permite ahorrar espacio.

Nombre	Detalles	Descripción
`zset-max-ziplist-entries`	Tipo: INTEGER Valores permitidos: 0- Valor predeterminado: 128	Determina la cantidad de memoria que se usa para los conjuntos ordenados. Los conjuntos ordenados con un número de elementos inferior al especificado se almacenan con una codificación especial que permite ahorrar espacio.
`zset-max-ziplist-value`	Tipo: INTEGER Valores permitidos: 0- Predeterminado: 64	Determina la cantidad de memoria que se usa para los conjuntos ordenados. Los conjuntos ordenados con entradas de tamaño inferior al número de bytes especificado se almacenan con una codificación especial que permite ahorrar espacio.
`tracking-table-max-keys`	Tipo: INTEGER Valores permitidos: 1-100000000 Valor predeterminado: 1000000	Para ayudar al almacenamiento en caché del lado del cliente, Redis admite el seguimiento de qué clientes han accedido a qué claves. Cuando se modifica la clave rastreada, se envían mensajes de invalidación a todos los clientes para notificarles que sus valores almacenados en caché ya no son válidos. Este valor permite especificar el límite superior de esta tabla.
`acllog-max-len`	Tipo: INTEGER Valores permitidos: 1-10000 Valor predeterminado: 128	El número máximo de entradas en el registro ACL.

Nombre	Detalles	Descripción
active-expire-effort	Tipo: INTEGER Valores permitidos: 1-10 Valor predeterminado: 1	Redis elimina las claves que han superado su periodo de vida por dos mecanismos. En uno, se accede a una clave y se encuentra que ha vencido. En el otro, un trabajo periódico muestra claves y hace que se venzan aquellas que han excedido su periodo de vida. Este parámetro define la cantidad de esfuerzo que Redis utiliza para vencer elementos en el trabajo periódico. El valor predeterminado de 1 intenta evitar tener más del 10 % de las claves vencidas que todavía se encuentran en la memoria. También intenta evitar consumir más del 25 % de la memoria total y agregar latencia al sistema. Puede aumentar este valor hasta 10 para aumentar la cantidad de esfuerzo invertido en las claves vencidas. La desventaja es mayor CPU y una latencia potencialmente mayor. Recomendamos un valor de 1 a menos que vea un uso elevado de memoria y pueda tolerar un aumento en la utilización de la CPU.
lazyfree-lazy-user-del	Tipo: STRING Valores permitidos: sí, no Valor predeterminado: no	Especifica si el comportamiento predeterminado del comando DEL actúa igual que UNLINK.
activedefrag	Tipo: STRING Valores permitidos: sí, no Valor predeterminado: no	Desfragmentación de memoria activa habilitada.

Nombre	Detalles	Descripción
`maxclients`	Tipo: INTEGER Valores permitidos: 65000 Predeterminado: 65000	Número máximo de clientes que pueden conectarse a la vez. No modificable.
`client-query-buffer-limit`	Tipo: INTEGER Valores permitidos: 1048576-1073741824 Predeterminado: 1073741824	Tamaño máximo de un búfer de consulta de cliente. Aplicación inmediata de los cambios.
`proto-max-bulk-len`	Tipo: INTEGER Valores permitidos: 1048576-536870912 Predeterminado: 536870912	Tamaño máximo de una sola solicitud de elemento. Aplicación inmediata de los cambios.

Parámetros específicos de tipo de nodo de MemoryDB

Aunque la mayoría de los parámetros tienen un único valor, algunos parámetros tienen distintos valores en función del tipo de nodo que se use. La tabla siguiente muestra el valor predeterminado `maxmemory` para cada tipo de nodo. El valor de `maxmemory` es el número máximo de bytes disponibles para el uso, los datos y otros usos en el nodo.

Tipo de nodo	Maxmemory
db.r7g.large	14037181030
db.r7g.xlarge	28261849702
db.r7g.2xlarge	56711183565

Tipo de nodo	Maxmemory
db.r7g.4xlarge	113609865216
db.r7g.8xlarge	225000375228
db.r7g.12xlarge	341206346547
db.r7g.16xlarge	450000750456
db.r6gd.xlarge	28261849702
db.r6gd.2xlarge	56711183565
db.r6gd.4xlarge	113609865216
db.r6gd.8xlarge	225000375228
db.r6g.large	14037181030
db.r6g.xlarge	28261849702
db.r6g.2xlarge	56711183565
db.r6g.4xlarge	113609865216
db.r6g.8xlarge	225000375228
db.r6g.12xlarge	341206346547
db.r6g.16xlarge	450000750456
db.t4g.small	1471026299
db.t4g.medium	3317862236

> **Note**
>
> Todos los tipos de instancia de MemoryDB se deben crear en una Amazon Virtual Private Cloud VPC.

Tutorial: Configuración de una función Lambda para acceder a MemoryDB en una Amazon VPC

En este tutorial, puede aprender a:

- Cree un clúster de MemoryDB en su Amazon Virtual Private Cloud (Amazon VPC) predeterminada en la región us-east-1.
- Cree una función Lambda para acceder al clúster. Al crear la función de Lambda debe proporcionar los ID de subred de Amazon VPC, así como un grupo de seguridad de VPC para que la función de Lambda pueda obtener acceso a los recursos de la VPC. A modo de ejemplo, en este tutorial, la función Lambda genera un UUID, lo escribe en el clúster y lo recupera del clúster.
- Invoque la función Lambda manualmente y compruebe que ha accedido al clúster de la VPC.
- Limpie la función Lambda, el clúster y el rol de IAM que se configuraron para este tutorial.

Temas

Paso 1: creación de un clúster

Para crear un clúster, siga estos pasos.

Temas

Paso 1.1: Crea un clúster

En este paso, crea un clúster en la Amazon VPC predeterminada de la región us-east-1 de su cuenta mediante la (AWS Command Line Interface CLI). Para obtener información sobre cómo crear un clúster mediante la consola o la API de MemoryDB, consulte. Paso 1: creación de un clúster

```
aws memorydb create-cluster --cluster-name cluster-01 --engine-version 7.0 --acl-name
 open-access \
--description "MemoryDB IAM auth application" \
--node-type db.r6g.large
```

Como puede ver, el valor del campo Estado es `CREATING`. MemoryDB puede tardar unos minutos en terminar de crear el clúster.

Paso 1.2: Copie el punto final del clúster

Compruebe que MemoryDB haya terminado de crear el clúster con el `describe-clusters` comando.

```
aws memorydb describe-clusters \
--cluster-name cluster-01
```

Copie la dirección del punto final del clúster que se muestra en el resultado. Necesitará esta dirección cuando cree el paquete de implementación para la función de Lambda.

Paso 1.3: Crear un rol de IAM

1. Cree un documento de política de confianza de IAM, como se muestra a continuación, para el rol que permita a la cuenta asumir el nuevo rol. Guarde la política en un archivo denominado trust-policy.json. Asegúrese de reemplazar el account_id 123456789012 en esta política por su account_id.

   ```
   {
   "Version": "2012-10-17",
       "Statement": [{
           "Effect": "Allow",
           "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
           "Action": "sts:AssumeRole"
       },
   ```

```
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }]
}
```

2. Cree un documento de política de IAM, como se muestra a continuación. Guarde la política en un archivo denominado policy.json. Asegúrate de reemplazar el account_id 123456789012 en esta política por tu account_id.

```
{
"Version": "2012-10-17",
  "Statement": [
    {
      "Effect" : "Allow",
      "Action" : [
        "memorydb:Connect"
      ],
      "Resource" : [
        "arn:aws:memorydb:us-east-1:123456789012:cluster/cluster-01",
        "arn:aws:memorydb:us-east-1:123456789012:user/iam-user-01"
      ]
    }
  ]
}
```

3. Crear un rol de IAM.

```
aws iam create-role \
--role-name "memorydb-iam-auth-app" \
--assume-role-policy-document file://trust-policy.json
```

4. Cree la política de IAM.

```
aws iam create-policy \
  --policy-name "memorydb-allow-all" \
  --policy-document file://policy.json
```

5. Adjunte la política de IAM al rol. Asegúrate de reemplazar el account_id 123456789012 en esta política (arn) por tu account_id.

```
aws iam attach-role-policy \
 --role-name "memorydb-iam-auth-app" \
 --policy-arn "arn:aws:iam::123456789012:policy/memorydb-allow-all"
```

Paso 1.4: Crear una lista de control de acceso (ACL)

1. Cree un nuevo usuario habilitado para IAM.

```
aws memorydb create-user \
  --user-name iam-user-01 \
--authentication-mode Type=iam \
--access-string "on ~* +@all"
```

2. Cree un nuevo usuario habilitado para IAM.

```
aws memorydb create-user \
  --user-name iam-user-01 \
--authentication-mode Type=iam \
--access-string "on ~* +@all"
```

3. Cree una ACL y adjúntela al clúster.

```
aws memorydb create-acl \
  --acl-name iam-acl-01 \
  --user-names iam-user-01

aws memorydb update-cluster \
  --cluster-name cluster-01 \
  --acl-name iam-acl-01
```

Paso 2: Crear una función de Lambda

Para crear una función Lambda, siga estos pasos.

Temas

Paso 2.1: creación del paquete de despliegue

En este tutorial, proporcionamos código de ejemplo en Python para su función Lambda.

Python

El siguiente ejemplo de código Python lee y escribe un elemento en el clúster de MemoryDB. Copie el código y guárdelo en un archivo con el nombre `app.py`. Asegúrese de reemplazar el `cluster_endpoint` valor del código por la dirección de punto final que copió en el paso 1.2.

```
from typing import Tuple, Union
from urllib.parse import ParseResult, urlencode, urlunparse

import botocore.session
import redis
from botocore.model import ServiceId
from botocore.signers import RequestSigner
from cachetools import TTLCache, cached
import uuid

class MemoryDBIAMProvider(redis.CredentialProvider):
    def __init__(self, user, cluster_name, region="us-east-1"):
        self.user = user
        self.cluster_name = cluster_name
        self.region = region

        session = botocore.session.get_session()
        self.request_signer = RequestSigner(
            ServiceId("memorydb"),
            self.region,
            "memorydb",
            "v4",
            session.get_credentials(),
            session.get_component("event_emitter"),
        )

    # Generated IAM tokens are valid for 15 minutes
    @cached(cache=TTLCache(maxsize=128, ttl=900))
    def get_credentials(self) -> Union[Tuple[str], Tuple[str, str]]:
        query_params = {"Action": "connect", "User": self.user}

        url = urlunparse(
            ParseResult(
```

```
                scheme="https",
                netloc=self.cluster_name,
                path="/",
                query=urlencode(query_params),
                params="",
                fragment="",
            )
        )
        signed_url = self.request_signer.generate_presigned_url(
            {"method": "GET", "url": url, "body": {}, "headers": {}, "context": {}},
            operation_name="connect",
            expires_in=900,
            region_name=self.region,
        )
        # RequestSigner only seems to work if the URL has a protocol, but
        # MemoryDB only accepts the URL without a protocol
        # So strip it off the signed URL before returning
        return (self.user, signed_url.removeprefix("https://"))

def lambda_handler(event, context):
    username = "iam-user-01" # replace with your user id
    cluster_name = "cluster-01" # replace with your cache name
    cluster_endpoint = "clustercfg.cluster-01.xxxxxx.memorydb.us-east-1.amazonaws.com"
 # replace with your cluster endpoint
    creds_provider = MemoryDBIAMProvider(user=username, cluster_name=cluster_name)
    redis_client = redis.Redis(host=cluster_endpoint, port=6379,
 credential_provider=creds_provider, ssl=True, ssl_cert_reqs="none")

    key='uuid'
    # create a random UUID - this will be the sample element we add to the cluster
    uuid_in = uuid.uuid4().hex
    redis_client.set(key, uuid_in)
    result = redis_client.get(key)
    decoded_result = result.decode("utf-8")
    # check the retrieved item matches the item added to the cluster and print
    # the results
    if decoded_result == uuid_in:
        print(f"Success: Inserted {uuid_in}. Fetched {decoded_result} from MemoryDB.")
    else:
        raise Exception(f"Bad value retrieved. Expected {uuid_in}, got
 {decoded_result}")

    return "Fetched value from MemoryDB"
```

Este código usa la `redis-py` biblioteca de Python para colocar elementos en el clúster y recuperarlos. Este código se utiliza `cachetools` para almacenar en caché los tokens de autenticación de IAM generados durante 15 minutos. Para crear un paquete de despliegue que contenga `redis-py` y `cachetools` lleve a cabo los siguientes pasos.

En el directorio del proyecto que contiene el archivo de código `app.py` fuente, cree un paquete de carpetas en el que instalar `cachetools` las bibliotecas `redis-py` y.

```
mkdir package
```

Instala `redis-py` y `cachetools` usa pip.

```
pip install --target ./package redis
pip install --target ./package cachetools
```

Cree un archivo.zip que contenga las bibliotecas `redis-py` y`cachetools`. En Linux y macOS, ejecuta el siguiente comando. En Windows, usa la utilidad zip que prefieras para crear un archivo.zip con las `cachetools` bibliotecas `redis-py` y en la raíz.

```
cd package
zip -r ../my_deployment_package.zip
```

Añada el código de función al archivo .zip. En Linux y macOS, ejecute el siguiente comando. En Windows, usa la utilidad zip que prefieras para agregar app.py a la raíz del archivo.zip.

```
cd package
zip -r ../my_deployment_package.zip
```

Paso 2.2: crear el rol de IAM (rol de ejecución)

Adjunte la política AWS administrada nombrada `AWSLambdaVPCAccessExecutionRole` al rol.

```
aws iam attach-role-policy \
 --role-name "memorydb-iam-auth-app" \
 --policy-arn "arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole"
```

Paso 2.3: cargar el paquete de despliegue (crear la función de Lambda)

En este paso, se crea la función Lambda (AccessMemoryDB) mediante el comando AWS CLI create-function.

Desde el directorio del proyecto que contiene el archivo.zip del paquete de despliegue, ejecute el siguiente comando de la `create-function` CLI de Lambda.

Para la opción de rol, utilice el ARN del rol de ejecución que creó en el paso 2.2. Para el vpc-config, introduzca listas separadas por comas de las subredes de la VPC predeterminada y el ID del grupo de seguridad de la VPC predeterminada. Puede encontrar estos valores en la Consola de Amazon VPC. Para buscar las subredes de su VPC predeterminadas, elija Sus VPC y, a continuación, elija la VPC predeterminada AWS de su cuenta. Para buscar el grupo de seguridad de esta VPC, vaya a Seguridad y elija Grupos de seguridad. Compruebe que ha seleccionado la región us-east-1.

```
aws lambda create-function \
--function-name AccessMemoryDB  \
--region us-east-1 \
--zip-file fileb://my_deployment_package.zip \
--role arn:aws:iam::123456789012:role/memorydb-iam-auth-app \
--handler app.lambda_handler \
--runtime python3.12  \
--timeout 30 \
--vpc-config SubnetIds=comma-separated-vpc-subnet-ids,SecurityGroupIds=default-
security-group-id
```

Paso 3: comprobación de la función de Lambda

En este paso, se invoca la función Lambda manualmente mediante el comando invoke. Cuando se ejecuta la función Lambda, genera un UUID y lo escribe en la ElastiCache caché que especificó en el código Lambda. A continuación, la función de Lambda recupera el elemento de la caché.

1. Invoque la función Lambda AccessMemory (DB) mediante AWS Lambda el comando invoke.

   ```
   aws lambda invoke \
   --function-name AccessMemoryDB  \
   --region us-east-1 \
   output.txt
   ```

2. Compruebe que la función de Lambda se ha ejecutado correctamente del modo siguiente:

   - Revise el archivo output.txt.
   - Verifique los resultados en los CloudWatch registros abriendo la CloudWatch consola y eligiendo el grupo de registros para su función (/aws/lambda/). AccessRedis El flujo de registro debería contener una salida similar a lo siguiente:

```
Success: Inserted 826e70c5f4d2478c8c18027125a3e01e. Fetched
 826e70c5f4d2478c8c18027125a3e01e from MemoryDB.
```

- Revise los resultados en la consola. AWS Lambda

Paso 4: limpiar (opcional)

Para limpiar, sigue estos pasos.

Temas

- Paso 4.1: Eliminar la función Lambda
- Paso 4.2: Eliminar el clúster de MemoryDB
- Paso 4.3: Eliminar la función y las políticas de IAM

Paso 4.1: Eliminar la función Lambda

```
aws lambda delete-function \
 --function-name AccessMemoryDB
```

Paso 4.2: Eliminar el clúster de MemoryDB

Eliminar el clúster.

```
aws memorydb delete-cluster \
 --cluster-name cluster-01
```

Elimine el usuario y la ACL.

```
aws memorydb delete-user \
 --user-id iam-user-01

aws memorydb delete-acl \
 --acl-name iam-acl-01
```

Paso 4.3: Eliminar la función y las políticas de IAM

```
aws iam detach-role-policy \
```

```
 --role-name "memorydb-iam-auth-app" \
 --policy-arn "arn:aws:iam::123456789012:policy/memorydb-allow-all"

aws iam detach-role-policy \
--role-name "memorydb-iam-auth-app" \
--policy-arn "arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole"

aws iam delete-role \
 --role-name "memorydb-iam-auth-app"

 aws iam delete-policy \
  --policy-arn "arn:aws:iam::123456789012:policy/memorydb-allow-all"
```

Búsqueda vectorial

Esta característica está en versión preliminar para MemoryDB para Redis y está sujeta a cambios.

La búsqueda vectorial de MemoryDB amplía la funcionalidad de MemoryDB. La búsqueda vectorial se puede utilizar junto con las funciones existentes de MemoryDB. La presencia de la búsqueda vectorial no afecta a las aplicaciones que no la utilizan. La vista previa de búsqueda vectorial está disponible a partir de la versión 7.1 de MemoryDB y en las siguientes regiones: Este de EE. UU. (Norte de Virginia y Ohio), Oeste de EE. UU. (Oregón), UE (Irlanda) y Asia Pacífico (Tokio).

La búsqueda vectorial de Amazon MemoryDB para Redis simplifica la arquitectura de la aplicación al mismo tiempo que ofrece una búsqueda vectorial de alta velocidad. La búsqueda vectorial de MemoryDB es ideal para casos de uso en los que el máximo rendimiento y la escalabilidad son los criterios de selección más importantes. Puede usar sus datos de MemoryDB existentes, o la API de Redis, para desarrollar casos de uso de machine learning e IA generativa, como la generación con recuperación aumentada, la detección de anomalías, la recuperación de documentos y las recomendaciones en tiempo real.

Temas

Información general de la búsqueda vectorial

Esta característica está en versión preliminar para MemoryDB para Redis y está sujeta a cambios.

La búsqueda vectorial está basada en la creación, el mantenimiento y el uso de índices. Cada operación de búsqueda vectorial especifica un índice único y su operación se limita a ese índice, es

decir, las operaciones de un índice no afectan las operaciones de ningún otro índice. A excepción de las operaciones de creación y destrucción de índices, se puede realizar cualquier cantidad de operaciones en cualquier índice en cualquier momento, lo que significa que, a nivel de clúster, pueden estar en ejecución varias operaciones en varios índices simultáneamente.

Los índices individuales son objetos con nombre que existen en un espacio de nombres único y separado de los demás espacios de nombres de Redis: claves, funciones, etc. Cada índice es conceptualmente similar a una tabla de base de datos convencional, dada su estructura en dos dimensiones: columnas y filas. Cada fila en la tabla corresponde a una clave de Redis. Cada columna del índice corresponde a un miembro o a una parte de esa clave. En este documento, los términos clave, fila y registro son idénticos y se usan indistintamente. Del mismo modo, los términos columna, campo, ruta y miembro son idénticos en esencia y también se usan indistintamente.

No existen comandos especiales para añadir, eliminar o modificar los datos indexados. Por el contrario, los comandos HASH o JSON existentes que modifican una clave que está en un índice también lo actualizan automáticamente.

Temas

Índices y el espacio de claves de Redis

Los índices se construyen y mantienen en un subconjunto del espacio de claves de Redis. Los índices múltiples pueden elegir subconjuntos disociados o superpuestos del espacio de claves de Redis sin limitación alguna. Durante la creación del índice se proporciona una lista de prefijos clave que definen el espacio de claves de cada índice. La lista de prefijos es opcional y, si se omite, todo el espacio de claves de Redis formará parte de ese índice. Los índices también están tipificados en el sentido de que solo incluyen las claves de tipo coincidente. Actualmente, solo se admiten los índices JSON y HASH. Un índice HASH solo indexa las claves HASH incluidas en su lista de prefijos y, de manera semejante, un índice JSON solo indexa las claves JSON incluidas en su lista de prefijos.

Las claves incluidas en la lista de prefijos del espacio de claves de un índice que no poseen el tipo designado se ignoran y no afectan a las operaciones de búsqueda.

Cuando un comando HASH o JSON modifica una clave que se encuentra dentro del espacio de claves de un índice, dicho índice se actualiza. Este proceso implica la extracción de los campos declarados para cada índice y la actualización del índice con el nuevo valor. El proceso de actualización ocurre en un subproceso en segundo plano, lo que significa que en última instancia los índices solo son coherentes con el contenido de su espacio de claves. Por lo tanto, la inserción o actualización de una clave no será visible en los resultados de búsqueda durante un breve período de tiempo. Durante los períodos en los que el sistema está sobrecargado o hay grandes cambios en los datos, el retraso en la visibilidad puede prolongarse.

La creación de un índice es un proceso de varios pasos. El primer paso es ejecutar el comando FT.CREATE que define el índice. Al ejecutarse correctamente el comando create, se inicia automáticamente el segundo paso: la reposición. El proceso de reposición se ejecuta en un subproceso en segundo plano y analiza el espacio de claves de Redis en busca de claves que estén dentro de la lista de prefijos del nuevo índice. Cada clave que se encuentra se agrega al índice. Finalmente, se analiza todo el espacio de claves y se completa el proceso de creación del índice. Tenga en cuenta que mientras el proceso de reposición está en marcha, se permiten las mutaciones de las claves indexadas, no hay restricciones y el proceso de reposición del índice no finalizará hasta que todas las claves estén indexadas correctamente. No se permiten las operaciones de consulta realizadas mientras se está rellenando un índice y se las finaliza con un error. La finalización del proceso de reposición se puede determinar a partir del resultado del comando `FT.INFO` para ese índice ('backfill_status').

El campo de índice escribe

Cada campo (columna) de un índice tiene un tipo específico que se declara durante la creación del índice y una ubicación dentro de una clave. En Claves HASH, la ubicación es el nombre del campo dentro del HASH. En Claves JSON, la ubicación es una descripción de la ruta JSON. Al modificar una clave, los datos asociados a los campos declarados se extraen, se convierten al tipo declarado y se almacenan en el índice. Si faltan los datos o no se pueden convertir correctamente al tipo declarado, ese campo se omite del índice. Hay cuatro tipos de campos, según se explica a continuación:

- Los campos numéricos contienen un solo número. En Campos JSON, se deben seguir las reglas numéricas de los números JSON. En HASH, se espera que el campo contenga el texto ASCII de un número escrito en el formato estándar para números de punto fijo o flotante.

Independientemente de la representación que contenga la clave, este campo se convierte en un número de punto flotante de 64 bits para almacenarlo en el índice. Los campos numéricos se pueden utilizar con el operador de búsqueda por rangos. Como los números subyacentes se almacenan en punto flotante con sus limitaciones de precisión, se aplican las reglas habituales sobre las comparaciones numéricas de números de punto flotante.

- Los campos de etiquetas contienen cero o más valores de etiqueta codificados como una sola cadena UTF-8. La cadena se analiza en valores de etiqueta mediante un carácter separador (el valor predeterminado es una coma, pero se puede anular) y se eliminan los espacios en blanco iniciales y finales. Se puede incluir cualquier número de valores de etiqueta en un único campo de etiqueta. Los campos de etiquetas se pueden usar para filtrar las consultas y determinar la equivalencia de los valores de las etiquetas mediante una comparación que distinga entre mayúsculas y minúsculas o que no distinga entre mayúsculas y minúsculas.
- Los campos de texto contienen una masa de bytes que no deben ser necesariamente compatibles con UTF-8. Los campos de texto se pueden usar para decorar los resultados de las consultas con valores significativos para la aplicación. Por ejemplo, una URL o el contenido de un documento, etc.
- Los campos vectoriales contienen un vector de números, también conocido como una incrustación. Los campos vectoriales admiten la búsqueda del k vecino más cercano (KNN) de vectores de tamaño fijo mediante un algoritmo y una métrica de distancia específicos. En Índices HASH, el campo debe contener todo el vector codificado en formato binario (IEEE 754 del tipo little-endian). En Claves JSON, la ruta debe hacer referencia a una matriz del tamaño correcto llena de números. Tenga en cuenta que cuando se utiliza una matriz JSON como campo vectorial, la representación interna de la matriz dentro de la clave JSON se convierte al formato exigido por el algoritmo seleccionado, lo que reduce el consumo y la precisión de memoria. Las operaciones de lectura posteriores con los comandos JSON darán como resultado el valor de precisión reducido.

Algoritmos de índice vectorial

Se proporcionan dos algoritmos de índice vectorial:

- FLAT: el algoritmo Flat es un procesamiento lineal de fuerza bruta de cada vector del índice, que da como resultado respuestas exactas dentro de los límites de la precisión de los cálculos de distancia. Debido al procesamiento lineal del índice, los tiempos de ejecución de este algoritmo pueden ser muy altos para índices grandes.
- HNSW (Jerarchical Navigable Small Worlds): el algoritmo HNSW es una alternativa que proporciona una aproximación de la respuesta correcta a cambio de tiempos de

ejecución sustancialmente más bajos. El algoritmo está controlado por tres parámetros, `M` `EF_CONSTRUCTION` y `EF_RUNTIME`. Los dos primeros parámetros se especifican en el momento de la creación del índice y no se pueden cambiar. El parámetro `EF_RUNTIME` tiene un valor predeterminado que se especifica al crear el índice, pero se puede anular posteriormente en cualquier operación de consulta individual. Estos tres parámetros interactúan para equilibrar el consumo de memoria y de CPU durante las operaciones de incorporación y consulta, así como para controlar la calidad de la aproximación de una búsqueda KNN exacta (conocida como relación de recuperación).

Ambos algoritmos de búsqueda vectorial (FLAT y HNSW) admiten un parámetro `INITIAL_CAP` opcional. Si se especifica, este parámetro asigna previamente memoria a los índices, lo que da como resultado una reducción de la sobrecarga de administración de la memoria y aumenta las tasas de incorporación vectorial.

Es posible que los algoritmos de búsqueda vectorial, como el HNSW, no gestionen de manera eficiente la eliminación o la sobrescritura de los vectores previamente insertados. El uso de estas operaciones puede provocar un consumo excesivo de memoria indexada o una calidad de la recuperación reducida. La reindexación es un método para restablecer el uso o la recuperación óptimos de la memoria.

Expresión de consulta de búsqueda vectorial

Los comandos FT.SEARCH y FT.AGGREGATE exigen una expresión de consulta. que es un parámetro de cadena única que se compone de uno o varios operadores. Cada operador utiliza un campo del índice para identificar un subconjunto de las claves del índice. Se pueden combinar varios operadores mediante combinadores booleanos y paréntesis para mejorar o restringir aún más el conjunto de claves (o conjunto de resultados) recopilado.

Comodín

El operador comodín, el asterisco ('*'), coincide con todas las claves del índice.

Rango numérico

El operador de rango numérico tiene la siguiente sintaxis:

```
<range-search> ::= '@' <numeric-field-name> ':' '[' <bound> <bound> ']'
<bound>  ::= <number> | '(' <number>
```

```
<number> ::= <integer> | <fixed-point> | <floating-point> | 'Inf' | '-Inf' | '+Inf'
```

El campo **<** numeric-field-name **>** debe ser un campo de tipo declarado. `NUMERIC` De forma predeterminada, el límite es inclusivo, pero se puede usar un paréntesis abierto inicial ['('] para hacer que un límite sea exclusivo. La búsqueda por rangos se puede convertir en una comparación relacional (**<**, **<=**, **> >** =) única mediante `Inf`, `+Inf` o `-Inf` como uno de los límites. Independientemente del formato numérico especificado (entero, punto fijo, punto flotante, infinito), el número se convierte en punto flotante de 64 bits para realizar comparaciones y, en consecuencia, reducir la precisión.

Example Ejemplos

```
@numeric-field:[0 10]                    // 0   <= <value> <= 10
@numeric-field:[(0 10]                   // 0   <  <value> <= 10
@numeric-field:[0 (10]                   // 0   <= <value> <  10
@numeric-field:[(0 (10]                  // 0   <  <value> <  10
@numeric-field:[1.5 (Inf]                // 1.5 <= value
```

Comparación de etiquetas

El operador de comparación de etiquetas tiene la siguiente sintaxis:

```
<tag-search> ::= '@' <tag-field-name> ':' '{' <tag> [ '|' <tag> ]* '}'
```

Si alguna de las etiquetas del operador coincide con alguna de las etiquetas del campo de etiquetas del registro, este se incluye en el conjunto de resultados. El campo diseñado por el `<tag-field-name>` debe ser un campo del índice declarado con el tipo `TAG`. Algunos ejemplos de una comparación de etiquetas son los siguientes:

```
@tag-field:{ atag }
@tag-field: { tag1 | tag2 }
```

Combinaciones booleanas

Los conjuntos de resultados de un operador numérico o de etiquetas se pueden combinar mediante la lógica booleana: y/o. Los paréntesis se pueden usar para agrupar operadores y/o cambiar el orden de la evaluación. La sintaxis de los operadores lógicos booleanos es la siguiente:

```
<expression> ::= <phrase> | <phrase> '|' <expression> | '(' <expression> ')'
```

```
<phrase> ::= <term> | <term> <phrase>
<term> ::= <range-search> | <tag-search> | '*'
```

Los términos múltiples combinados en una frase son anexados con “y”. Las frases múltiples combinadas con la barra vertical ('|') se relacionan con “o”.

Búsqueda vectorial

El operador de búsqueda vectorial realiza una búsqueda de k vecinos más cercanos de un índice de campo vectorial. La sintaxis de una búsqueda vectorial es la siguiente:

```
<vector-search> ::= <expression> '=>[KNN' <k> '@' <vector-field-name> '$' <parameter-
name> <modifiers> ']'
<modifiers> ::= [ 'EF_RUNTIME' <integer> ] [ 'AS' <distance-field-name>]
```

La búsqueda vectorial solo se aplica a los vectores que cumplen con `<expression>`, que pueden ser cualquier combinación de los operadores definidos anteriormente: comodín, búsqueda por rango, búsqueda por etiquetas y/o combinaciones booleanas de los mismos.

- <k> es un número entero que especifica el número de vectores vecinos más cercanos que se van a devolver.
- `<vector-field-name>` debe especificar un campo de tipo `VECTOR` declarado.
- El campo `<parameter-name>` especifica una de las entradas de la tabla `PARAM` del comando `FT.SEARCH` o `FT.AGGREGATE`. Este parámetro es el valor vectorial de referencia para los cálculos de distancia. El valor del vector está codificado en el valor `PARAM` del formato binario IEEE 754 del tipo little-endian (con la misma codificación que para un campo vectorial HASH)
- Para los índices vectoriales de tipo HNSW, la cláusula `EF_RUNTIME` opcional se puede utilizar para anular el valor predeterminado del parámetro `EF_RUNTIME` que se estableció cuando se creó el índice.
- La `<distance-field-name>` opcional proporciona un nombre de campo para que el conjunto de resultados contenga la distancia calculada entre el vector de referencia y la clave ubicada.

Comando INFO

La búsqueda vectorial amplía el comando INFO de Redis con varias secciones adicionales de estadísticas y contadores. Al solicitar la recuperación de la sección `SEARCH`, se recuperarán todas las secciones siguientes:

Sección de **`search_memory`**

Nombre	Descripción
search_used_memory_bytes	Número de bytes de memoria consumidos en todas las estructuras de datos de búsqueda
search_used_memory_human	Versión legible por seres humanos de lo anterior

Sección de **`search_index_stats`**

Nombre	Descripción
search_number_of_indexes	Número de índices creados
search_num_fulltext_indexes	Número de campos no vectoriales en todos los índices
search_num_vector_indexes	Número de campos vectoriales en todos los índices
search_num_hash_indexes	Número de índices en las claves de tipo HASH
search_num_json_indexes	Número de índices en las claves de tipo JSON
search_total_indexed_keys	Número total de claves en todos los índices
search_total_indexed_vectors	Número total de vectores en todos los índices
search_total_indexed_hash_keys	Número total de claves de tipo HASH en todos los índices
search_total_indexed_json_keys	Número total de claves de tipo JSON en todos los índices
search_total_index_size	Bytes utilizados por todos los índices

Nombre	Descripción
search_total_fulltext_index_size	Bytes utilizados por estructuras de índices no vectoriales
search_total_vector_index_size	Bytes utilizados por estructuras de índices vectoriales
search_max_index_lag_ms	Retraso de incorporación durante la última actualización del lote de incorporación

Sección de **`search_ingestion`**

Nombre	Descripción
search_background_indexing_status	Estado de la incorporación. `NO_ACTIVITY` significa inactivo. Otros valores indican que hay claves en proceso de incorporación.
search_ingestion_paused	A menos que se reinicie, siempre debe ser “no”.

Sección de **`search_backfill`**

> **Note**
>
> Algunos de los campos documentados en esta sección solo están visibles cuando hay una reposición en curso.

Nombre	Descripción
search_num_active_backfills	Número de actividades de reposición actuales
search_backfills_paused	Excepto cuando se agote la memoria, siempre debe ser “no”.

Nombre	Descripción
search_highest_backfill_progress_percentage	% de finalización (0-100) de la reposición más completada
search_lowest_backfill_progress_percentage	% de finalización (0-100) de la reposición menos completada

Sección de **`search_query`**

Nombre	Descripción
search_num_active_queries	Número de comandos `FT.SEARCH` y `FT.AGGREGATE` actualmente en curso

Seguridad de búsqueda vectorial

Los mecanismos de seguridad ACL (listas de control de acceso) de Redis para el acceso a los comandos y a los datos se han ampliado para controlar la función de búsqueda. El control ACL de los comandos de búsqueda individuales es totalmente compatible. Se proporciona una nueva categoría de ACL, `@search`, y muchas de las categorías existentes (`@fast`, `@read`, `@write`, etc.) se actualizan para incluir los nuevos comandos. Los comandos de búsqueda no modifican los datos clave, lo que significa que se conserva la maquinaria ACL existente para el acceso de escritura. La presencia de un índice no modifica las reglas de acceso para las operaciones HASH y JSON; se sigue aplicando el control de acceso normal a nivel de clave a estos comandos.

El acceso de los comandos de búsqueda con un índice también se controla mediante la ACL de Redis. Las comprobaciones de acceso se realizan a nivel de índice completo, no al nivel de la clave. Esto significa que el acceso a un índice se garantiza a un usuario solo si ese usuario tiene permiso para acceder a todas las claves posibles de la lista de prefijos del espacio de claves de ese índice. En otras palabras, el contenido real de un índice no controla el acceso. Más bien, es el contenido teórico de un índice, tal como se define en la lista de prefijos, el que se utiliza para el control de seguridad. Puede ser sencillo crear una situación en la que un usuario tenga acceso de lectura o escritura a una clave, pero no tenga acceso a un índice que contenga esa clave. Tenga en cuenta que solo se requiere acceso de lectura al espacio de claves para crear o usar un índice; no se tiene en cuenta la presencia o ausencia del acceso de escritura.

Para obtener más información, consulte Autenticación de usuarios con listas de control de acceso (ACL).

Características y límites de la búsqueda vectorial

Esta característica está en versión preliminar para MemoryDB para Redis y está sujeta a cambios.

Disponibilidad de búsqueda vectorial

La configuración MemoryDB, que permite la búsqueda vectorial, es compatible con los tipos de nodos R6g, R7g y T4g y está disponible en las siguientes AWS regiones: EE.UU. Este (Norte de Virginia), EE.UU. Este (Ohio), EE.UU. Oeste (Oregón), Asia-Pacífico (Tokio) y Europa (Irlanda).

Restricciones paramétricas

En la siguiente tabla se muestran los límites de varios elementos de búsqueda vectorial en la vista previa:

Elemento	Valor máximo
Cantidad de dimensiones de un vector	32768
Cantidad de índices que se pueden crear	10
Cantidad de campos de un índice	50
Cantidad de operaciones de reposición simultáneas del índice FT.CREATE	1
Cláusulas de tiempo de espera FT.SEARCH y FT.AGGREGATE (milisegundos)	60000
Cantidad de etapas de canalización en el comando FT.AGGREGATE	32
Cantidad de campos de la cláusula FT.AGGREGATE LOAD	1024

Elemento	Valor máximo
Cantidad de campos de la cláusula FT.AGGREGATE GROUPBY	16
Cantidad de campos de la cláusula FT.AGGREGATE SORTBY	16
Cantidad de parámetros en la cláusula FT.AGGREGATE PARAM	32
Parámetro HNSW M	512
Parámetro HNSW EF_CONSTRUCTION	4096
Parámetro HNSW EF_RUNTIME	4096

Límites de escalado

La búsqueda vectorial de MemoryDB está limitada actualmente a una única partición y no se admite el escalado horizontal. La búsqueda vectorial admite el escalado vertical y de réplica.

Restricciones operativas

Persistencia y reposición de índices

La vista previa de la búsqueda vectorial conserva la definición de los índices, pero no su contenido. Por lo tanto, cualquier solicitud o evento operativo que provoque el inicio o el reinicio de un nodo requiere que todos los índices se reconstruyan a partir de sus datos clave de definición y de origen. El proceso de reconstrucción se inicia automáticamente una vez que se han restaurado todos los datos; no es necesaria ninguna acción por parte del usuario para iniciarlo. La reconstrucción se realiza como una operación de reposición tan pronto como se restauran los datos. Esto equivale funcionalmente a que el sistema ejecute automáticamente un comando FT.CREATE para cada índice definido. Tenga en cuenta que el nodo estará disponible para las operaciones de la aplicación tan pronto como se restauren los datos, pero probablemente antes de que se complete la reposición del índice, lo que significa que las aplicaciones volverán a estar visibles para las aplicaciones. Por ejemplo, es posible que se rechacen los comandos de búsqueda que utilicen índices de reposición.

Para obtener más información sobre la reposición, consulte Información general de la búsqueda vectorial.

La finalización de la reposición del índice no se sincroniza entre una reposición principal y una réplica. Esta falta de sincronización puede pasar a ser visible para las aplicaciones de forma inesperada, por lo que se recomienda que las aplicaciones verifiquen que esté finalizada la reposición en las principales y todas las réplicas antes de iniciar las operaciones de búsqueda.

Importación y exportación de instantáneas y migración en tiempo real

La presencia de índices de búsqueda en un archivo RDB limita la transportabilidad compatible de esos datos. El formato de los índices definido por la edición de vista previa solo lo entiende otro clúster de ediciones de vista previa. Por lo tanto, los archivos RDB con índices de búsqueda solo se pueden transferir entre clústeres de MemoryDB habilitados para la vista previa o ser utilizados por estos.

Sin embargo, los archivos RDB que no contienen índices no están restringidos de esta manera. Por lo tanto, los datos de un clúster de vista previa se pueden exportar a clústeres que no sean de vista previa mediante la eliminación de los índices antes de la exportación.

Consumo de memoria

La implementación actual de índices vectoriales consume aproximadamente el doble de la cantidad de memoria que la que consumirá la implementación de disponibilidad general.

Falta de memoria durante la reposición

Al igual que las operaciones de escritura de Redis, el relleno de índices está sujeto a limitaciones. out-of-memory Si la memoria de Redis se llena mientras hay una reposición en curso, todas las reposiciones se pausan. Si queda memoria disponible, se reanuda el proceso de reposición. También es posible eliminar e indexar cuando la reposición está en pausa por falta de memoria.

Transacciones

Los comandos `FT.CREATE`, `FT.DROPINDEX`, `FT.ALIASADD`, `FT.ALIASDEL` y `FT.ALIASUPDATE` no se pueden ejecutar en un contexto transaccional, es decir, no dentro de un bloque MULTI/EXEC ni dentro de un script LUA o FUNCTION.

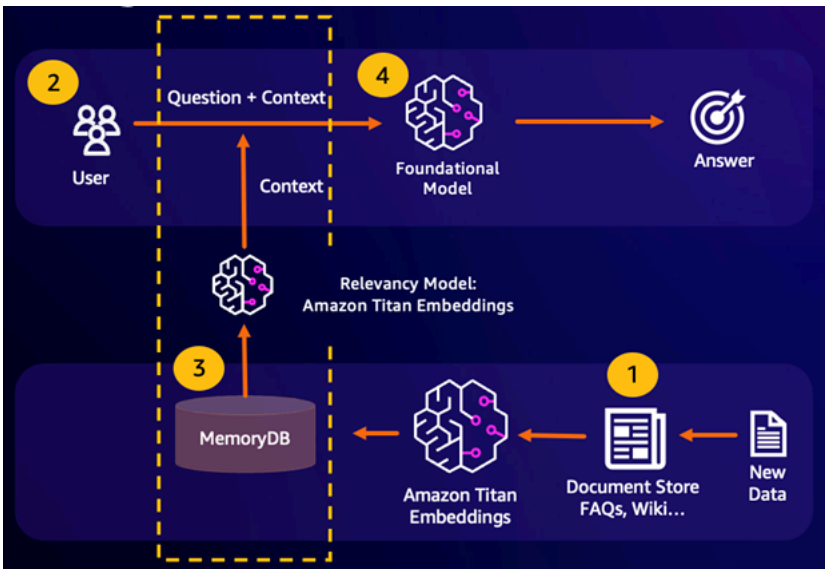
Casos de uso

Esta característica está en versión preliminar para MemoryDB para Redis y está sujeta a cambios.

A continuación se presentan algunos casos de uso de búsqueda vectorial.

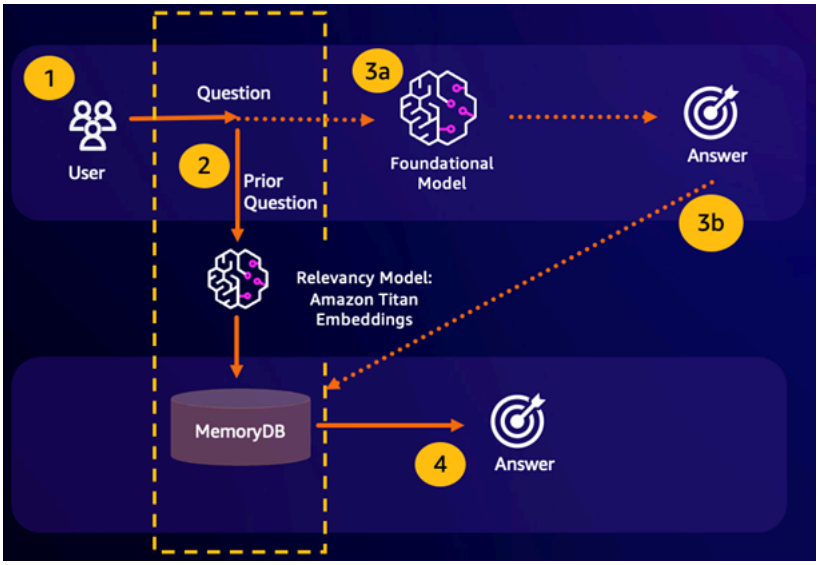
Generación aumentada de recuperación (RAG)

La generación aumentada de recuperación (RAG) aprovecha la búsqueda vectorial para recuperar pasajes relevantes de un gran corpus de datos para aumentar modelo de lenguaje grande (LLM). En concreto, un codificador incrusta el contexto de entrada y la consulta de búsqueda en vectores y, a continuación, utiliza la búsqueda aproximada de vecino más cercano para encontrar pasajes semánticamente similares. Estos pasajes recuperados se concatenan con el contexto original para proporcionar información adicional pertinente al LLM y devolver una respuesta más precisa al usuario.



Memoria búfer del modelo fundacional (FM)

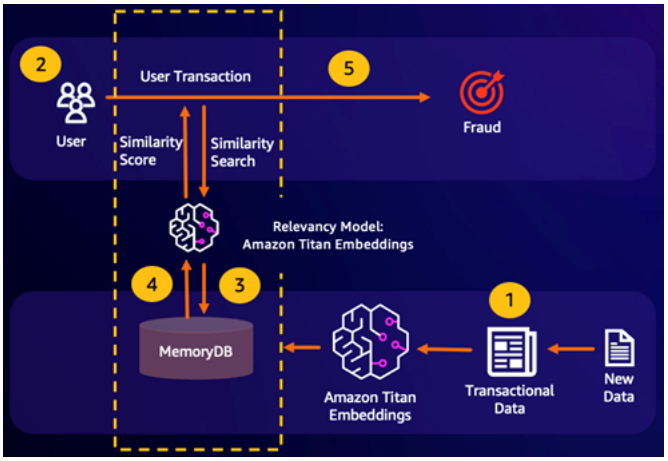
La memoria búfer del modelo fundacional (FM) es un proceso para reducir los costos computacionales al almacenar los resultados anteriores de la FM. Al reutilizar los resultados anteriores de inferencias previas en lugar de volver a calcularlos, la memoria intermedia de FM reduce la cantidad de cálculo necesaria durante la inferencia a través de los FM. Esta memoria búfer de FM permite que los modelos lingüísticos de gran tamaño respondan más rápido y con costos más bajos debido a los cargos de servicio del FM.



- Resultado de la búsqueda semántica: si la consulta de un cliente es semánticamente similar en función de una puntuación de similitud definida con una pregunta anterior, la memoria intermedia del FM (MemoryDB) devolverá la respuesta a la pregunta anterior en el paso 4 y no llamará al FM en los pasos 3. Esto evitará la latencia del modelo fundacional (FM) y los costos incurridos, lo que permitirá al cliente disfrutar de una experiencia más rápida.
- Fallo en la búsqueda semántica: si la consulta de un cliente no es semánticamente similar en función de una puntuación de similitud definida con respecto a una consulta anterior, el cliente llamará al FM para responderle en el paso 3a. La respuesta generada por el FM se almacenará luego como vector en MemoryDB para consultas futuras (paso 3b) a fin de minimizar los costos del FM en preguntas semánticamente similares. En este flujo, no se invocaría el paso 4 porque no había una pregunta semánticamente similar para la consulta original.

Detección de fraudes

La detección de fraudes, una forma de detección de anomalías, representa las transacciones válidas como vectores al tiempo que compara las representaciones vectoriales de las nuevas transacciones netas. El fraude se detecta cuando estas nuevas transacciones netas tienen una baja similitud con los vectores que representan los datos transaccionales válidos. Esto permite detectar el fraude al modelar el comportamiento normal, en lugar de intentar predecir todas las posibles instancias de fraude. MemoryDB permite a las organizaciones hacerlo en períodos de alto rendimiento, con un mínimo de falsos positivos y una latencia de milisegundos de un solo dígito.



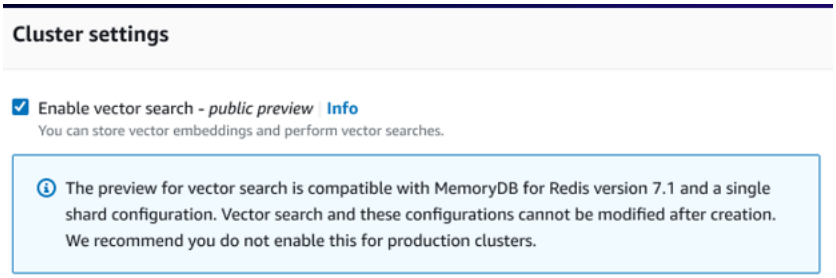
Otros casos de uso

- Los motores de recomendación pueden encontrar productos o contenidos similares para los usuarios al representar los elementos como vectores. Los vectores se crean mediante el análisis de atributos y patrones. Según los patrones y atributos del usuario, se pueden recomendar nuevos elementos invisibles a los usuarios al encontrar los vectores más similares que ya hayan sido puntuados positivamente y alineados con el usuario.
- Los motores de búsqueda de documentos representan los documentos de texto como vectores densos de números que capturan su significado semántico. En el momento de la búsqueda, el motor convierte una consulta de búsqueda en un vector y encuentra los documentos con los vectores más similares a la consulta mediante una búsqueda aproximada del vecino más cercano. Este enfoque de similitud vectorial permite coincidir los documentos en función del significado en lugar de simplemente hacer coincidir las palabras clave.

Usando el AWS Management Console

Esta característica está en versión preliminar para MemoryDB para Redis y está sujeta a cambios.

Para crear un clúster que permita la búsqueda vectorial en la consola, debe habilitar la búsqueda vectorial en la configuración del clúster. La búsqueda vectorial está disponible con versión 7.1 de MemoryDB para Redis y para una configuración de partición única.



Para obtener más información sobre el uso de la búsqueda vectorial con AWS Management Console, consulteCreación de un clúster (consola).

Uso del AWS Command Line Interface

Esta característica está en versión preliminar para MemoryDB para Redis y está sujeta a cambios.

Para crear un clúster de MemoryDB habilitado para la búsqueda vectorial, puede utilizar el comando crear-clúster de MemoryDB al pasar un grupo de parámetros inmutable `default.memorydb-redis7.search.preview` para habilitar el modo de vista previa de las capacidades de búsqueda vectorial.

```
aws memorydb create-cluster \
  --cluster-name <value> \
  --node-type <value> \
  --engine redis \
  --engine-version 7.1 \
  --num-shards 1 \
  --acl-name <value> \
  --parameter-group-name default.memorydb-redis7.search.preview
```

Comandos de búsqueda vectorial

A continuación se muestra una lista de comandos compatibles para la búsqueda vectorial.

Temas

- FT.CREATE
- FT.SEARCH

- FT.AGGREGATE
- FT.DROPINDEX
- FT.INFO
- FT._LIST
- FT.ALIASADD
- FT.ALIASDEL
- FT.ALIASUPDATE
- FT._ALIASLIST
- FT.CONFIG GET
- FT.CONFIG HELP
- FT.CONFIG SET
- FT.PROFILE
- FT.EXPLAIN
- FT.EXPLAINCLI

FT.CREATE

Crea un índice e inicia la reposición de ese índice. Para obtener más información, consulte la descripción general de la búsqueda vectorial para obtener más detalles sobre la construcción del índice.

Sintaxis

```
FT.CREATE <index-name>
ON HASH | JSON
[PREFIX <count> <prefix1> [<prefix2>...]]
SCHEMA
(<field-identifier> [AS <alias>]
  NUMERIC
| TAG [SEPARATOR <sep>] [CASESENSITIVE]
| TEXT
| VECTOR [HNSW|FLAT] <attr_count> [<attribute_name> <attribute_value>])

)+
```

Esquema

- Identificador del campo:
  - En Claves hash, el identificador de campo es Un nombre de campo.
  - En Claves JSON, el identificador de campo es Una ruta JSON.

  Para obtener más información, consulte El campo de índice escribe.

- Tipos de campo:
  - ETIQUETA: Para obtener más información, consulte Etiquetas.
  - NUMÉRICO: el campo contiene un número.
  - TEXTO: El campo contiene cualquier bloque de datos.
  - VECTOR: campo vectorial que admite la búsqueda vectorial.
    - Algoritmo: puede ser HNSW (mundo pequeño navegable jerárquicamente) o FLAT (fuerza bruta).
    - `attr_count`: cantidad de atributos que se transferirán como configuración del algoritmo, que incluye tanto los nombres como los valores.
    - `{attribute_name} {attribute_value}`: pares clave/valor específicos del algoritmo que definen la configuración del índice.

      Para el algoritmo FLAT, los atributos son:

      Obligatorio
      - DIM: la cantidad de dimensiones del vector.
      - DISTANCE_METRIC: puede ser uno de los siguientes: [L2 | IP | COSINE].
      - TYPE: tipo de vector. El único tipo admitido es `FLOAT32`.

      Opcional:
      - INITIAL_CAP: capacidad vectorial inicial del índice que afecta al tamaño de asignación de memoria del índice.

      Para el algoritmo HNSW, los atributos son:

      Obligatorio
      - TYPE: tipo de vector. El único tipo admitido es `FLOAT32`.
      - DIM: dimensión vectorial, especificada como un entero positivo. Máximo: 32768
      - DISTANCE_METRIC: puede ser uno de los siguientes: [L2 | IP | COSINE].

Opcional:

- INITIAL_CAP: capacidad vectorial inicial del índice que afecta al tamaño de asignación de memoria del índice. El valor predeterminado es 1024.
- M: cantidad máxima de bordes salientes permitidos para cada nodo del gráfico en cada capa. En la capa cero, el número máximo de bordes salientes será de 2 millones. El valor predeterminado es 16 y el máximo es 512.
- EF_CONSTRUCTION: controla la cantidad de vectores examinados durante la construcción del índice. Los valores más altos de este parámetro mejorarán la tasa de recuperación a costa de prolongar los tiempos de creación del índice. El valor predeterminado es 200. El valor máximo es 4096.
- EF_RUNTIME: controla la cantidad de vectores examinados durante las operaciones de consulta. Los valores más altos de este parámetro darán una tasa de recuperación mejorada a costa de tiempos de consulta prolongados. El valor de este parámetro se puede anular según cada consulta. El valor predeterminado es 10. El valor máximo es 4096.

Devolución

Devuelve un mensaje de OK de cadena simple o una respuesta de error.

Ejemplos

> **Note**
>
> En el siguiente ejemplo, se utilizan argumentos nativos de redis-cli, como eliminar las comillas y los valores de escape de los datos, antes de enviarlos a Redis. Para usar otros clientes de lenguajes de programación (Python, Ruby, C#, etc.), siga las reglas de manejo de esos entornos para tratamiento de cadenas y datos binarios. Para obtener más información sobre los clientes compatibles, consulte Herramientas sobre las que basarse AWS

Example 1: Crear algunos índices

Cree un índice para vectores de tamaño 2

```
FT.CREATE hash_idx1 ON HASH PREFIX 1 hash: SCHEMA vec AS VEC VECTOR HNSW 6 DIM 2 TYPE
 FLOAT32 DISTANCE_METRIC L2
OK
```

Cree un índice JSON de 6 dimensiones mediante el algoritmo HNSW:

```
FT.CREATE json_idx1 ON JSON PREFIX 1 json: SCHEMA $.vec AS VEC VECTOR HNSW 6 DIM 6 TYPE
 FLOAT32 DISTANCE_METRIC L2
OK
```

Example Ejemplo 2: rellenar algunos datos

Los siguientes comandos están formateados para que se puedan ejecutar como argumentos en el programa de terminal redis-cli. Los desarrolladores que utilicen clientes de lenguajes de programación (como Python, Ruby, C#, etc.) deberán seguir las reglas de manejo de su entorno para tratar cadenas y datos binarios.

Crear algunos datos hash y json:

```
HSET hash:0 vec "\x00\x00\x00\x00\x00\x00\x00\x00"
HSET hash:1 vec "\x00\x00\x00\x00\x00\x00\x80\xbf"
JSON.SET json:0 . '{"vec":[1,2,3,4,5,6]}'
JSON.SET json:1 . '{"vec":[10,20,30,40,50,60]}'
JSON.SET json:2 . '{"vec":[1.1,1.2,1.3,1.4,1.5,1.6]}'
```

Tenga en cuenta lo siguiente:

- Las claves de los datos HASH y JSON tienen los prefijos de sus definiciones de índice.
- Los vectores se encuentran en las rutas apropiadas de las definiciones del índice.
- Los vectores HASH se ingresan como datos hexadecimales, mientras que los datos JSON se ingresan como números.
- Los vectores tienen las longitudes adecuadas, las entradas del vector HASH bidimensional tienen dos valores flotantes de datos hexadecimales y las entradas vectoriales json de seis dimensiones tienen seis números.

Example Ejemplo 3: Eliminar y volver a crear un índice

```
FT.DROPINDEX json_idx1
OK

FT.CREATE json_idx1 ON JSON PREFIX 1 json: SCHEMA $.vec AS VEC VECTOR FLAT 6 DIM 6 TYPE
 FLOAT32 DISTANCE_METRIC L2
```

```
OK
```

Tenga en cuenta que el nuevo índice JSON usa el algoritmo `FLAT` en lugar del algoritmo `HNSW`. Tenga en cuenta también que volverá a indexar los datos JSON existentes:

```
FT.SEARCH json_idx1 "*=>[KNN 100 @VEC $query_vec]" PARAMS 2 query_vec
 "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
 DIALECT 2
1) (integer) 3
2) "json:2"
3) 1) "__VEC_score"
   2) "11.11"
   3) "$"
   4) "[{\"vec\":[1.1, 1.2, 1.3, 1.4, 1.5, 1.6]}]"
4) "json:0"
5) 1) "__VEC_score"
   2) "91"
   3) "$"
   4) "[{\"vec\":[1.0, 2.0, 3.0, 4.0, 5.0, 6.0]}]"
6) "json:1"
7) 1) "__VEC_score"
   2) "9100"
   3) "$"
   4) "[{\"vec\":[10.0, 20.0, 30.0, 40.0, 50.0, 60.0]}]"
```

FT.SEARCH

Utiliza la expresión de consulta proporcionada para localizar las claves dentro de un índice. Una vez localizadas, se puede devolver el recuento o el contenido de los campos indexados dentro de esas claves. Para obtener más información, consulte Expresión de consulta de búsqueda vectorial.

Para crear datos para utilizarlos en estos ejemplos, consulte el comando FT.CREATE.

Sintaxis

```
FT.SEARCH <index-name> <query>
[RETURN <token_count> (<field-identifier> [AS <alias>])+]
[TIMEOUT timeout]
[PARAMS <count> <name> <value> [<name> <value>]]
[LIMIT <offset> <count>]
[COUNT]
```

- RETURN: esta cláusula identifica qué campos de una clave se devuelven. La cláusula AS opcional de cada campo anula el nombre del campo en el resultado. Solo se pueden especificar los campos que se han declarado para este índice.
- LIMIT: <offset><count>: Esta cláusula proporciona capacidad de paginación, ya que solo se devuelven las claves que cumplen los valores de compensación y recuento. Si se omite esta cláusula, el valor predeterminado es “LIMIT 0 10”, es decir, solo se devolverá un máximo de 10 claves.
- PARAMS: dos veces la cantidad de pares de valores clave. Se puede hacer referencia a los pares clave/valor de los parámetros desde la expresión de consulta. Para obtener más información, consulte Expresión de consulta de búsqueda vectorial.
- COUNT: esta cláusula impide que se devuelva el contenido de las claves, solo se devuelve la cantidad de claves. Es un alias para “LIMIT 0 0”.

Devolución

Devuelve una matriz o una respuesta de error.

- Si la operación se completa correctamente, devuelve una matriz. El primer elemento es la cantidad total de claves que coinciden con la consulta. Los elementos restantes son pares de nombre de clave y la lista de campos. La lista de campos es otra matriz que comprende pares de nombres y valores de campo.
- Si el índice está en proceso de reposición, el comando devuelve inmediatamente una respuesta de error.
- Si se agota el tiempo de espera, el comando devuelve una respuesta de error.

Ejemplo: haz algunas búsquedas

> **Note**
>
> En el siguiente ejemplo, se utilizan argumentos nativos de redis-cli, como eliminar las comillas y los valores de escape de los datos, antes de enviarlos a Redis. Para usar otros clientes de lenguajes de programación (Python, Ruby, C#, etc.), siga las reglas de manejo de esos entornos para tratamiento de cadenas y datos binarios. Para obtener más información sobre los clientes compatibles, consulte Herramientas sobre las que basarse AWS

Una búsqueda de hash

```
FT.SEARCH hash_idx1 "*=>[KNN 2 @VEC $query_vec]" PARAMS 2 query_vec
 "\x00\x00\x00\x00\x00\x00\x00\x00" DIALECT 2
1) (integer) 2
2) "hash:0"
3) 1) "__VEC_score"
   2) "0"
   3) "vec"
   4) "\x00\x00\x00\x00\x00\x00\x00\x00"
4) "hash:1"
5) 1) "__VEC_score"
   2) "1"
   3) "vec"
   4) "\x00\x00\x00\x00\x00\x00\x80\xbf"
```

Esto produce dos resultados, ordenados por su puntuación, que es la distancia desde el vector de consulta (introducido como hexadecimal).

Búsquedas en JSON

```
FT.SEARCH json_idx1 "*=>[KNN 2 @VEC $query_vec]" PARAMS 2 query_vec
 "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
 DIALECT 2
1) (integer) 2
2) "json:2"
3) 1) "__VEC_score"
   2) "11.11"
   3) "$"
   4) "[{\"vec\":[1.1, 1.2, 1.3, 1.4, 1.5, 1.6]}]"
4) "json:0"
5) 1) "__VEC_score"
   2) "91"
   3) "$"
   4) "[{\"vec\":[1.0, 2.0, 3.0, 4.0, 5.0, 6.0]}]"
```

Esto produce los dos resultados más cercanos, ordenados por su puntuación. Tenga en cuenta que los valores vectoriales JSON se convierten en flotantes y el vector de consulta sigue constando de datos vectoriales. Tenga en cuenta también que, dado que el parámetro KNN es 2, solo hay dos resultados. Un valor mayor devolverá más resultados:

```
FT.SEARCH json_idx1 "*=>[KNN 100 @VEC $query_vec]" PARAMS 2 query_vec
 "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
 DIALECT 2
1) (integer) 3
2) "json:2"
3) 1) "__VEC_score"
   2) "11.11"
   3) "$"
   4) "[{\"vec\":[1.1, 1.2, 1.3, 1.4, 1.5, 1.6]}]"
4) "json:0"
5) 1) "__VEC_score"
   2) "91"
   3) "$"
   4) "[{\"vec\":[1.0, 2.0, 3.0, 4.0, 5.0, 6.0]}]"
6) "json:1"
7) 1) "__VEC_score"
   2) "9100"
   3) "$"
   4) "[{\"vec\":[10.0, 20.0, 30.0, 40.0, 50.0, 60.0]}]"
```

FT.AGGREGATE

Como superconjunto del comando FT.SEARCH, permite un procesamiento adicional sustancial de las claves seleccionadas por la expresión de consulta.

Sintaxis

```
FT.AGGREGATE index query
  [LOAD * | [count field [field ...]]]
  [TIMEOUT timeout]
  [PARAMS count name value [name value ...]]
  [FILTER expression]
  [LIMIT offset num]
  [GROUPBY count property [property ...] [REDUCE function count arg [arg ...] [AS name]
 [REDUCE function count arg [arg ...] [AS name] ...]] ...]]
  [SORTBY count [ property ASC | DESC [property ASC | DESC ...]] [MAX num]]
  [APPLY expression AS name]
```

- Las cláusulas FILTER, LIMIT, GROUPBY, SORTBY y APPLY se pueden repetir varias veces en cualquier orden y mezclarse libremente. Se aplican en el orden especificado y el resultado de una cláusula alimenta la entrada de la siguiente cláusula.

- En la sintaxis anterior, una “propiedad” es un campo declarado en el comando FT.CREATE para este índice O el resultado de una cláusula APPLY o función REDUCE anterior.
- La cláusula LOAD se limita a cargar campos que se han declarado en el índice. “LOAD *” cargará todos los campos declarados en el índice.
- Se admiten las siguientes funciones reductoras: COUNT, COUNT_DISTINCTISH, SUM, MIN, MAX, AVG, STDDEV, QUANTILE, TOLIST, FIRST_VALUE y RANDOM_SAMPLE. Para obtener más información, consulte Agregaciones.
- LIMIT <offset><count>: conserva los registros que comienzan en <offset>y continúan hasta <count>, todos los demás registros se descartan.
- PARAMS: dos veces la cantidad de pares de valores clave. Se puede hacer referencia a los pares clave/valor de los parámetros desde la expresión de consulta. Para obtener más información, consulte Expresión de consulta de búsqueda vectorial.

Devolución

Devuelve una matriz o una respuesta de error.

- Si la operación se completa correctamente, devuelve una matriz. El primer elemento es un número entero sin ningún significado particular (debe ignorarse). Los elementos restantes son los resultados obtenidos en la última etapa. Cada elemento es una matriz de pares de nombre y valor de campo.
- Si el índice está en proceso de reposición, el comando devuelve inmediatamente una respuesta de error.
- Si se agota el tiempo de espera, el comando devuelve una respuesta de error.

FT.DROPINDEX

Elimine un índice. Se eliminan la definición del índice y el contenido asociado. Las claves de Redis no se ven afectadas.

Sintaxis

```
FT.DROPINDEX <index-name>
```

Devolución

Devuelve un mensaje de OK de cadena simple o una respuesta de error.

FT.INFO

Sintaxis

```
FT.INFO <index-name>
```

El resultado de la página FT.INFO es una matriz de pares de valores clave, tal como se describe en la siguiente tabla:

Clave	Tipo de valor	Descripción
index_name	cadena	Nombre del índice
creation_timestamp	integer	Marca temporal de la hora de creación de estilo Unix
key_type	cadena	HASH o JSON
key_prefixes	matriz de cadenas	Prefijos clave para este índice
campos	matriz de información de campo	Campos de este índice
space_usage	integer	Bytes de memoria utilizados por este índice
fullext_space_usage	integer	Bytes de memoria utilizados por campos no vectoriales
vector_space_usage	integer	Bytes de memoria utilizados por campos vectoriales
num_docs	integer	Número de claves que contiene actualmente el índice
num_indexed_vectors	integer	Número de vectores que contiene actualmente el índice
current_lag	integer	Retraso reciente de la incorporación (milisegundos)

Clave	Tipo de valor	Descripción
backfill_status	cadena	Una de las siguientes opciones: completada InProgres, pausada o fallida

La tabla siguiente describe información para cada campo:

Clave	Tipo de valor	Descripción
identifier	cadena	nombre del campo
field_name	cadena	Nombre del miembro del HASH o ruta JSON
type	cadena	uno de los siguiente s: numérico, de etiqueta, de texto o vectorial
option	cadena	ignore

Si el campo es del tipo Vector, habrá información adicional en función del algoritmo.

Para el algoritmo HNSW:

Clave	Tipo de valor	Descripción
algoritmo	cadena	HNSW
data_type	cadena	FLOAT32
distance_metric	cadena	uno de los siguientes: L2, IP o Cosine
initial_capacity	integer	Tamaño inicial del índice de campo vectorial

Clave	Tipo de valor	Descripción
current_capacity	integer	Tamaño actual del índice de campo vectorial
maximum_edges	integer	Parámetro M en el momento de la creación
ef_construction	integer	Parámetro EF_CONSTR UCTION en el momento de la creación
ef_runtime	integer	Parámetro EF_RUNTIME en el momento de la creación

Para el algoritmo FLAT:

Clave	Tipo de valor	Descripción
algoritmo	cadena	FLAT
data_type	cadena	FLOAT32
distance_metric	cadena	uno de los siguientes: L2, IP o Cosine
initial_capacity	integer	Tamaño inicial del índice de campo vectorial
current_capacity	integer	Tamaño actual del índice de campo vectorial

FT._LIST

Enumera todos los índices.

Sintaxis

```
FT._LIST
```

Devolución

Devuelve una matriz de nombres de índice

FT.ALIASADD

Añada un alias para un índice. El nuevo nombre de alias se puede usar en cualquier lugar donde se requiera un nombre de índice.

Sintaxis

```
FT.ALIASADD <alias> <index-name>
```

Devolución

Devuelve un mensaje de OK de cadena simple o una respuesta de error.

FT.ALIASDEL

Elimine un alias existente para un índice.

Sintaxis

```
FT.ALIASDEL <alias>
```

Devolución

Devuelve un mensaje de OK de cadena simple o una respuesta de error.

FT.ALIASUPDATE

Actualice un alias existente para que apunte a un índice físico diferente. Este comando solo afecta a las futuras referencias sobre el alias. Este comando no afecta a las operaciones actualmente en curso (FT.SEARCH, FT.AGGREGATE).

Sintaxis

```
FT.ALIASUPDATE <alias> <index>
```

Devolución

Devuelve un mensaje de OK de cadena simple o una respuesta de error.

FT._ALIASLIST

Enumera los alias del índice.

Sintaxis

```
FT._ALIASLIST
```

Devolución

Devuelve una matriz del tamaño del número de alias actuales. Cada elemento de la matriz es el par alias-índice.

FT.CONFIG GET

Devuelve el valor del parámetro TIMEOUT.

Sintaxis

```
FT.CONFIG GET [* | <timeout>]
```

Devolución

Devuelve el valor del parámetro TIMEOUT.

FT.CONFIG HELP

Recupera información sobre el parámetro TIMEOUT.

Sintaxis

```
FT.CONFIG HELP [* | <timeout>]
```

Devolución

Devuelve información sobre el parámetro TIMEOUT

FT.CONFIG SET

Establece el parámetro TIMEOUT. El valor predeterminado es 10 000 milisegundos.

> **Note**
>
> Los nombres de los parámetros configurables no distinguen entre mayúsculas y minúsculas.

Sintaxis

```
FT.CONFIG SET <timeout> <value>
```

Devolución

Devuelve el valor del ajuste TIMEOUT.

FT.PROFILE

Ejecuta una consulta y devuelve la información de perfil sobre esa consulta.

Sintaxis

```
FT.PROFILE

<index>
SEARCH | AGGREGATE
[LIMITED]
QUERY <query ....>
```

Devolución

Matriz de dos elementos. El primer elemento es el resultado del comando `FT.SEARCH` o `FT.AGGREGATE` que se perfiló. El segundo elemento es una matriz de información de rendimiento y creación de perfiles.

FT.EXPLAIN

Analiza una consulta y devuelve información sobre cómo se analizó esa consulta.

Sintaxis

```
FT.EXPLAIN <index> <query>
```

Devolución

Una cadena que contiene los resultados analizados.

FT.EXPLAINCLI

Igual que el comando FT.EXPLAIN, excepto que los resultados se muestran en un formato diferente, más útil con redis-cli.

Sintaxis

```
FT.EXPLAINCLI <index> <query>
```

Devolución

Una cadena que contiene los resultados analizados.

Seguridad en MemoryDB para Redis

La seguridad en la nube de AWS es la mayor prioridad. Como cliente de AWS, se beneficia de una arquitectura de red y un centro de datos que se han diseñado para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

La seguridad es una responsabilidad compartida entre AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta los servicios de AWS en la nube de AWS. AWS también proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWSProgramas de conformidad de . Para obtener información sobre los programas de conformidad que se aplican a MemoryDB, consulte Servicios de AWS en el ámbito del programa de conformidad.
- Seguridad en la nube: su responsabilidad viene determinada por el servicio de AWS que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza MemoryDB para Redis. Muestra cómo configurar MemoryDB para satisfacer sus objetivos de seguridad y conformidad. También puede aprender a utilizar otros servicios de AWS que lo ayuden a monitorear y proteger sus recursos de MemoryDB.

Contenido

Protección de datos en MemoryDB para Redis

El modelo de responsabilidad compartida de AWS se aplica a la protección de datos en Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta toda la Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog Modelo de responsabilidad compartida y GDPR de AWS en el Blog de seguridad de AWS.

Con fines de protección de datos, recomendamos proteger las credenciales de la cuenta de Cuenta de AWS y configurar cuentas de usuario individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se conceden a cada usuario los permisos necesarios para cumplir con sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos de AWS. reSe recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad del usuario con AWS CloudTrail.
- Utilice las soluciones de cifrado de AWS, junto con todos los controles de seguridad predeterminados dentro de los Servicios de AWS.
- Utilice servicios de seguridad gestionados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados FIPS 140-2 al acceder a AWS a través de una interfaz de la línea de comandos o una API, utilice un punto de conexión de FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

Se recomienda encarecidamente no ingresar nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Incluye las situaciones en las que debe trabajar con otros Servicios de AWS a través de la consola, la API, la AWS CLI o los SDK de AWS. Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos

encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Seguridad de los datos en MemoryDB para Redis

Para ayudar a proteger los datos, MemoryDB para Redis y Amazon EC2 incluyen mecanismos que impiden el acceso no autorizado a los datos del servidor.

MemoryDB también proporciona funciones de cifrado para datos en clústeres:

- El cifrado en tránsito cifra los datos mientras se mueven de un lugar a otro; por ejemplo, entre los nodos del clúster o entre el clúster y la aplicación.
- El cifrado en reposo cifra el registro de transacciones y los datos en el disco durante las operaciones de instantáneas.

También puede usar Autenticación de usuarios con listas de control de acceso (ACL) para controlar el acceso de los usuarios al clúster.

Temas

- Cifrado en reposo en MemoryDB
- Cifrado en tránsito (TLS) de MemoryDB
- Autenticación de usuarios con listas de control de acceso (ACL)
- Autenticación con IAM

Cifrado en reposo en MemoryDB

Para ayudarle a mantener los datos protegidos, MemoryDB para Redis y Amazon S3 cuentan con diferentes formas que permiten restringir el acceso a los datos de sus clústeres. Para obtener más información, consulte MemoryDB y Amazon VPC y Administración de identidades y accesos en MemoryDB para Redis.

El cifrado en reposo de MemoryDB siempre está activado para aumentar la seguridad de la información al cifrar los datos persistentes. Encripta los siguientes aspectos:

- Datos del registro de transacciones
- Disco durante las operaciones de sincronización, instantáneas o intercambio
- Instantáneas almacenadas en Amazon S3

MemoryDB ofrece cifrado en reposo predeterminado (servicio administrado), así como capacidad para usar sus propias claves maestras simétricas del cliente administradas por el cliente en AWS Key Management Service (KMS).

Los datos almacenados en los SSD (unidades de estado sólido) en clústeres habilitados para la organización de datos en niveles siempre se cifran de forma predeterminada.

Para obtener más información sobre el cifrado en tránsito, consulte Cifrado en tránsito (TLS) de MemoryDB.

Temas

Uso de claves administradas por el cliente desde AWS KMS

MemoryDB admite las claves maestras simétricas administradas por el cliente (clave de KMS) para el cifrado en reposo. Las claves de KMS administradas por el cliente son claves de cifrado que crea, posee y administra en la cuenta de AWS. Para obtener más información, consulte Claves raíz del cliente en la Guía para desarrolladores de AWS Key Management Service. Las claves deben crearse en AWS KMS para poder utilizarlas con MemoryDB.

Para obtener más información sobre la creación de claves maestras de AWS KMS, consulte Creación de claves en la Guía para desarrolladores de AWS Key Management Service.

MemoryDB permite la integración con AWS KMS. Para obtener más información, consulte Uso de concesiones en la Guía para desarrolladores de AWS Key Management Service. No se requieren acciones del cliente para habilitar la integración de MemoryDB con AWS KMS.

La clave de condición `kms:ViaService` limita el uso de una clave de AWS KMS a determinadas solicitudes de servicios de AWS. Para utilizar `kms:ViaService` con MemoryDB, incluya ambos nombres de ViaService en el valor de clave de condición: `memorydb.amazon_region.amazonaws.com`. Para obtener más información, consulte kms:ViaService.

Puede utilizar AWS CloudTrail para realizar un seguimiento de las solicitudes que MemoryDB para Redis envía a AWS Key Management Service en su nombre. Todas las llamadas a la API a AWS Key Management Service relacionadas con claves administradas por el cliente tienen los registros de CloudTrail correspondientes. También puede ver las concesiones que crea MemoryDB llamando a la API de KMS ListGrants.

Una vez que se cifra un clúster mediante la clave administrada por el cliente, todas las instantáneas para el clúster se cifran de la siguiente manera:

- Las instantáneas diarias automáticas se cifran mediante la clave administrada por el cliente asociada con el clúster.
- La instantánea final creada cuando se elimina el clúster también se cifra mediante la clave administrada por el cliente asociada con el clúster.
- Las instantáneas creadas de forma manual se cifran de manera predeterminada para utilizar la clave de KMS asociada con el clúster. Puede anular esto al elegir otra clave administrada por el cliente.
- Al copiar una instantánea se utiliza de forma predeterminada una clave administrada por el cliente asociada a la instantánea de origen. Puede anular esto al elegir otra clave administrada por el cliente.

> **Note**
>
> - Las claves administradas por el cliente no se pueden utilizar cuando se exportan instantáneas al bucket de Amazon S3 seleccionado. Sin embargo, todas las instantáneas exportadas a Amazon S3 se cifran mediante el cifrado del lado del servidor. Puede optar por copiar el archivo de instantánea en un objeto de S3 nuevo y cifrarlo mediante una clave de KMS administrada por el cliente, copiar el archivo a otro bucket de S3 que se haya

configurado con el cifrado predeterminado mediante una clave de KMS o cambiar una opción de cifrado en el propio archivo.

- También puede utilizar claves administradas por el cliente a fin de cifrar instantáneas creadas de forma manual que no utilicen claves administradas por el cliente para el cifrado. Con esta opción, el archivo de instantánea almacenado en Amazon S3 se cifra mediante una clave de KMS, aunque los datos no se cifren en el clúster original.

La restauración desde una instantánea le permite elegir entre las opciones de cifrado disponibles, similares a las opciones de cifrado disponibles cuando se crea un nuevo clúster.

- Si elimina la clave o deshabilita la clave y revoca las concesiones para la clave que utilizó para cifrar un clúster, el clúster se vuelve irrecuperable. En otras palabras, no se puede modificar ni recuperar después de un error de hardware. AWS KMS solo elimina las claves maestras después de un periodo de espera de al menos siete días. Después de eliminar la clave, puede utilizar una clave administrada por el cliente diferente para crear una instantánea con fines de archivo.
- La rotación automática de claves conserva las propiedades de las claves maestras de AWS KMS, por lo que la rotación no tiene efecto sobre la capacidad de acceder a los datos de MemoryDB. Los clústeres de MemoryDB no admiten la rotación de claves manual, lo que implica la creación de una nueva clave maestra y la actualización de cualquier referencia a la antigua clave. Para obtener más información, consulte Rotación de claves del cliente en la Guía para desarrolladores de AWS Key Management Service.
- El cifrado de un clúster de MemoryDB mediante la clave de KMS requiere una concesión por clúster. Esa concesión se utiliza a lo largo de la vida útil del clúster. Además, se utiliza una concesión por instantánea durante la creación de la instantánea. Dicha concesión se retira una vez que se crea la instantánea.
- Para obtener más información sobre las cuotas y las concesiones de AWS KMS, consulte Cuotas en la Guía para desarrolladores de AWS Key Management Service.

Véase también

- Cifrado en tránsito (TLS) de MemoryDB
- MemoryDB y Amazon VPC
- Administración de identidades y accesos en MemoryDB para Redis

Cifrado en tránsito (TLS) de MemoryDB

Para ayudar a proteger los datos, MemoryDB para Redis y Amazon EC2 incluyen mecanismos que impiden el acceso no autorizado a los datos del servidor. Al contar con una funcionalidad de cifrado en tránsito, MemoryDB le brinda una herramienta que puede utilizar para ayudar a proteger los datos cuando se trasladan de una ubicación a otra. Por ejemplo, puede mover datos de un nodo principal a un nodo de réplica de lectura de un clúster o entre el clúster y la aplicación.

Temas

Información general sobre el cifrado en tránsito

El cifrado en tránsito de MemoryDB para Redis es una característica que permite reforzar la seguridad de los datos en sus momentos más vulnerables: cuando se trasladan de una ubicación a otra.

El cifrado en tránsito de MemoryDB implementa las siguientes características:

- Conexiones cifradas: las conexiones del servidor y el cliente se cifran con la seguridad de la capa de transporte (TLS).
- Replicación cifrada: se cifran los datos que se trasladan entre un nodo principal y los nodos de réplica.
- Autenticación de servidores: los clientes pueden autenticar que se encuentran conectados al servidor correcto.

A partir del 20 de julio de 2023, TLS 1.2 es la versión mínima admitida para los clústeres nuevos y existentes. Utilice este enlace para obtener más información sobre TLS 1.2 en AWS.

Para obtener más información acerca de la conexión a los clústeres de MemoryDB, consulte Conexión a los nodos de MemoryDB mediante redis-cli.

Véase también

- Cifrado en reposo en MemoryDB
- Autenticación de usuarios con listas de control de acceso (ACL)

- MemoryDB y Amazon VPC
- Administración de identidades y accesos en MemoryDB para Redis

Autenticación de usuarios con listas de control de acceso (ACL)

Puede autenticar usuarios con listas de control de acceso (ACL).

Las ACL le permiten controlar el acceso a los clústeres agrupando a los usuarios. Estas listas de control de acceso se han diseñado como una forma de organizar el acceso a los clústeres.

Con las ACL, puede crear usuarios y asignarles permisos específicos mediante una cadena de acceso, como se describe en la sección a continuación. Asigne los usuarios a listas de control de acceso alineadas con un rol específico (administradores, recursos humanos) que luego se implementan en uno o más clústeres de MemoryDB. De esta manera, puede establecer límites de seguridad entre clientes que utilicen el mismo clúster o clústeres de MemoryDB e impedir que los clientes obtengan acceso a los datos de los demás.

Las ACL se han diseñado para apoyar la introducción de ACL de Redis en Redis 6. Cuando utiliza ACL con el clúster de MemoryDB, existen algunas limitaciones que debe tener en cuenta:

- No puede especificar contraseñas en una cadena de acceso. Establece contraseñas con CreateUsero UpdateUserllamadas.
- Para los derechos de usuario, pasa on y `off` como parte de la cadena de acceso. Si no se especifica ninguno en la cadena de acceso, se asigna `off` al usuario y no tiene derechos de acceso al clúster.
- No se pueden utilizar comandos prohibidos. Si especifica un comando prohibido, se generará una excepción. Para ver una lista de dichos comandos, consulte Comandos de Redis restringidos.
- No puede utilizar el comando `reset` como parte de una cadena de acceso. Las contraseñas se especifican con parámetros de la API y MemoryDB administra las contraseñas. Por lo tanto, no puede utilizar `reset` porque eliminará todas las contraseñas de un usuario.
- Redis 6 presenta el comando ACL LIST. Este comando devuelve una lista de usuarios junto con las reglas de ACL aplicadas a cada usuario. MemoryDB admite el comando `ACL LIST`, pero no incluye soporte para hash de contraseña como lo hace Redis. Con MemoryDB, puede utilizar la DescribeUsersoperación para obtener información similar, incluidas las reglas contenidas en la cadena de acceso. Sin embargo, DescribeUsersno recupera la contraseña de un usuario.

Otros comandos de solo lectura admitidos por MemoryDB incluyen ACL WHOAMI, ACL USERS y ACL CAT. MemoryDB no admite otros comandos ACL basados en escritura.

A continuación, se describe con más detalle el uso de ACL con MemoryDB.

Temas

- Especificación de permisos mediante una cadena de acceso
- Capacidades de la búsqueda vectorial
- Aplicar las ACL a un clúster para MemoryDB

Especificación de permisos mediante una cadena de acceso

Para especificar los permisos de un clúster de MemoryDB, debe crear una cadena de acceso y asignarla a un usuario mediante la AWS CLI tecla o. AWS Management Console

Las cadenas de acceso se definen como una lista de reglas delimitadas por espacios que se aplican al usuario. Definen qué comandos puede ejecutar un usuario y qué claves puede operar. Para ejecutar un comando, un usuario debe tener acceso al comando que se ejecuta y a todas las claves a las que accede el comando. Las reglas se aplican de izquierda a derecha de forma acumulativa y se puede utilizar una cadena más simple en lugar de la proporcionada si hay redundancias en la cadena proporcionada.

Para obtener más información sobre la sintaxis de las reglas de ACL, consulte ACL.

En el siguiente ejemplo, la cadena de acceso representa un usuario activo con acceso a todas las claves y comandos disponibles.

```
on ~* &* +@all
```

La sintaxis de la cadena de acceso se desglosa de la siguiente manera:

- on: el usuario es un usuario activo.
- ~*: se brinda acceso a todas las claves disponibles.
- &*— Se permite el acceso a todos los canales de pubsub.
- +@all: se brinda acceso a todos los comandos disponibles.

La configuración anterior es la menos restrictiva. Puede modificar esta configuración para hacerla más segura.

En el siguiente ejemplo, la cadena de acceso representa a un usuario con acceso restringido al acceso de lectura en claves que comienzan por el espacio de claves “app::”

```
on ~app::* -@all +@read
```

Puede refinar aún más estos permisos al enumerar comandos a los que el usuario tiene acceso:

`+`*`command1`*: el acceso del usuario a los comandos se encuentra limitado a *`command1`*.

`+@category`: el acceso del usuario a los comandos se encuentra limitado a la categoría de comandos.

Para obtener información sobre cómo asignar una cadena de acceso a un usuario, consulte Creación de usuarios y listas de control de acceso con la consola y la CLI.

Si va a migrar una carga de trabajo existente a MemoryDB, puede recuperar la cadena de acceso mediante una llamada a `ACL LIST`, que excluya el usuario y cualquier hash de contraseña.

Capacidades de la búsqueda vectorial

> **Note**
>
> Esta característica está en versión preliminar para MemoryDB para Redis y está sujeta a cambios.

En Búsqueda vectorial, todos los comandos de la búsqueda pertenecen a la categoría `@search`, y las categorías existentes `@read`, `@write`, `@fast` y `@slow` se actualizan para incluir los comandos de la búsqueda. Si un usuario no tiene acceso a una categoría, entonces no tiene acceso a ningún comando de la categoría. Por ejemplo, si el usuario no tiene acceso a `@search`, entonces no puede ejecutar ningún comando relacionado con la búsqueda.

En la siguiente tabla se indica la asignación de los comandos JSON a las categorías apropiadas.

Comandos de VSS	@read	@write	@fast	@slow
FT.CREATE		Y	Y	
FT.DROPINDEX		Y	Y	
FT.LIST	Y			Y
FT.INFO	Y		Y	
FT.SEARCH	Y			Y
FT.AGGREGATE	Y			Y
FT.PROFILE	Y			Y
FT.ALIASADD		Y	Y	
FT.ALIASDEL		Y	Y	
FT.ALIASUPDATE		Y	Y	
FT._ALIASLIST	Y			Y
FT.EXPLAIN	Y		Y	
FT.EXPLAINCLI	Y		Y	

Comandos de VSS	@read	@write	@fast	@slow
`FT.CONFIG`	Y		Y	

Aplicar las ACL a un clúster para MemoryDB

Para usar las ACL de MemoryDB, siga estos pasos:

1. Cree uno o más usuarios.
2. Cree una ACL y agregue usuarios a la lista.
3. Asigne la ACL a un clúster.

Estos pasos se describen en la siguiente tabla.

Temas

- Creación de usuarios y listas de control de acceso con la consola y la CLI
- Administración de listas de control de acceso con la consola y la CLI
- Asignación de listas de control de acceso a clústeres

Creación de usuarios y listas de control de acceso con la consola y la CLI

La información de usuario para los usuarios de las ACL es un nombre de usuario y, opcionalmente, una contraseña y una cadena de acceso. La cadena de acceso proporciona el nivel de permisos en las claves y comandos. El nombre de usuario es exclusivo del usuario y es lo que se pasa al motor.

Asegúrese de que los permisos de usuario que proporcione tengan sentido con el propósito previsto de la ACL. Por ejemplo, si crea una ACL denominada `Administrators`, cualquier usuario que agregue a ese grupo debe tener su cadena de acceso establecida en el acceso completo a las claves y comandos. Para los usuarios de una ACL de `e-commerce`, puede establecer las cadenas de acceso en acceso de solo lectura.

MemoryDB configura automáticamente un usuario predeterminado por cuenta con un nombre de usuario `"default"`. No se asociará a ningún clúster a menos que se añada explícitamente a una ACL. No puede modificar ni eliminar este usuario. Este usuario se ha diseñado para ser compatible

con el comportamiento predeterminado de las versiones anteriores de Redis y tiene una cadena de acceso que permite llamar a todos los comandos y acceder a todas las claves.

Se creará una ACL inmutable de “acceso abierto” para cada cuenta que contenga el usuario predeterminado. Esta es la única ACL a la que el usuario predeterminado puede pertenecer. Al crear un clúster, es preciso asociarlo con una ACL. Si bien tiene la opción de aplicar la ACL de “acceso abierto” con el usuario predeterminado, le recomendamos encarecidamente que cree una ACL con usuarios que tengan permisos restringidos a sus necesidades empresariales.

Los clústeres que no tienen habilitada la TLS deben usar la ACL de “acceso abierto” para proporcionar una autenticación abierta.

Las ACL se pueden crear sin usuarios. Una ACL vacía no tendría acceso a un clúster y solo se puede asociar a clústeres habilitados para TLS.

Al crear un usuario, puede configurar hasta dos contraseñas. Al modificar una contraseña, se mantienen todas las conexiones existentes a los clústeres.

En concreto, tenga en cuenta estas restricciones de contraseña del usuario al utilizar ACL con MemoryDB:

- Las contraseñas deben tener entre 16 y 128 caracteres imprimibles.
- No se admiten los siguientes caracteres no alfanuméricos: `, "" / @`.

Administración de usuarios con la consola y la CLI

Creación de usuarios (consola)

Para crear usuarios con la consola

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación izquierdo, elija Usuarios.
3. Elija Crear usuario
4. En la página Crear usuario, introduzca un nombre.

   Las restricciones para la asignación de nombres de clúster son las siguientes:

   - Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.

- Deben comenzar por una letra.
- No pueden contener dos guiones consecutivos.
- No pueden terminar con un guion.

5. En Contraseñas, puede introducir hasta dos contraseñas.
6. En Cadena de acceso, introduzca una cadena de acceso. La cadena de acceso establece el nivel de permisos para qué claves y comandos se permite al usuario.
7. En el caso de las etiquetas, si lo desea, puede aplicar etiquetas para buscar y filtrar a sus usuarios o realizar un seguimiento de sus costes. AWS
8. Seleccione Crear.

Crear un usuario mediante AWS CLI

Para crear un usuario mediante la CLI

- Utilice el comando create-user para crear un usuario.

  Para Linux, macOS o Unix:

```
aws memorydb create-user \
  --user-name user-name-1 \
  --access-string "~objects:* ~items:* ~public:*" \
  --authentication-mode \
        Passwords="abc",Type=password
```

  Para Windows:

```
aws memorydb create-user ^
  --user-name user-name-1 ^
  --access-string "~objects:* ~items:* ~public:*" ^
  --authentication-mode \
        Passwords="abc",Type=password
```

Modificación de un usuario (consola)

Para modificar usuarios con la consola

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación izquierdo, elija Usuarios.
3. Elija el botón de opción situado junto al usuario que desea modificar y luego elija Acciones -> Modificar
4. Si desea modificar una contraseña, pulse el botón de opción Modificar contraseñas. Tenga en cuenta que si tiene dos contraseñas, debe introducir ambas al modificar una de ellas.
5. Si va a actualizar la cadena de acceso, introduzca la nueva.
6. Elija Modificar.

Modificar un usuario mediante AWS CLI

Para modificar un usuario mediante la CLI

1. Utilice el comando update-user para modificar un usuario.
2. Cuando se modifica un usuario, se actualizan las listas de control de acceso asociadas al usuario, junto con los clústeres asociados a la ACL. Se mantienen todas las conexiones existentes. A continuación se muestran algunos ejemplos.

   Para Linux, macOS o Unix:

   ```
   aws memorydb update-user \
     --user-name user-name-1 \
     --access-string "~objects:* ~items:* ~public:*"
   ```

   Para Windows:

   ```
   aws memorydb update-user ^
     --user-name user-name-1 ^
     --access-string "~objects:* ~items:* ~public:*"
   ```

Visualización de detalles de los usuarios (consola)

Para ver los detalles del usuario en la consola

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación izquierdo, elija Usuarios.
3. Elija el usuario en Nombre de usuario o utilice el cuadro de búsqueda para encontrarlo.
4. En Configuración de usuario, puede revisar la cadena de acceso del usuario, el recuento de contraseñas, el estado y el nombre del recurso de Amazon (ARN).
5. En las listas de control de acceso (ACL), puede revisar la ACL a la que pertenece el usuario.
6. En Etiquetas, puede revisar cualquier etiqueta asociada al usuario.

Visualización de los detalles del usuario mediante el AWS CLI

Utilice el comando describe-users para ver los detalles de un usuario.

```
aws memorydb describe-users \
  --user-name my-user-name
```

Eliminación de un usuario (consola)

Para eliminar usuarios con la consola

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación izquierdo, elija Usuarios.
3. Elija el botón de opción situado junto al usuario que desea modificar y luego elija Acciones -> Eliminar
4. Para confirmar, en el cuadro de texto de confirmación, introduzca `delete` y, a continuación, elija Eliminar.
5. Para cancelar, elija Cancelar.

Eliminar un usuario mediante el AWS CLI

Para eliminar un usuario mediante la CLI

- Utilice el comando delete-user para eliminar un usuario.

  La cuenta se borra y elimina de todas las listas de control de acceso a las que pertenezca. A continuación, se muestra un ejemplo.

  Para Linux, macOS o Unix:

```
aws memorydb delete-user \
  --user-name user-name-2
```

  Para Windows:

```
aws memorydb delete-user ^
  --user-name user-name-2
```

Administración de listas de control de acceso con la consola y la CLI

Puede crear listas de control de acceso para organizar y controlar el acceso de los usuarios a uno o más clústeres, como se muestra a continuación.

Use el siguiente procedimiento para administrar las listas de control de acceso mediante la consola.

Creación de una lista de control de acceso (ACL) (consola)

Para crear una lista de control de acceso mediante la consola

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación izquierdo, elija Listas de control de acceso (ACL).
3. Seleccione Crear ACL.
4. En la página Crear lista de control de acceso (ACL), introduzca un nombre ACL.

   Las restricciones para la asignación de nombres de clúster son las siguientes:

   - Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.

- Deben comenzar por una letra.
- No pueden contener dos guiones consecutivos.
- No pueden terminar con un guion.

5. En Usuarios seleccionados, realice una de las siguientes acciones:

   a. Para crear un nuevo usuario, seleccione Crear usuario

   b. Para agregar usuarios, elija Administrar y, a continuación, seleccione los usuarios en el cuadro de diálogo Administrar usuarios y, a continuación, seleccione Elegir.

6. En el caso de las etiquetas, si lo desea, puede aplicar etiquetas para buscar y filtrar sus ACL o realizar un seguimiento de sus costes. AWS

7. Seleccione Crear.

Creación de una lista de control de acceso (ACL) mediante AWS CLI

Utilice el siguiente procedimiento para crear una lista de control de acceso mediante la CLI.

Para crear una nueva ACL y agregar un usuario mediante la CLI

- Utilice el comando create-acl para crear una ACL.

  Para Linux, macOS o Unix:

```
aws memorydb create-acl \
  --acl-name "new-acl-1" \
  --user-names "user-name-1" "user-name-2"
```

  Para Windows:

```
aws memorydb create-acl ^
  --acl-name "new-acl-1" ^
  --user-names "user-name-1" "user-name-2"
```

Modificación de una lista de control de acceso (ACL) (consola)

Para modificar una lista de control de acceso mediante la consola

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación izquierdo, elija Listas de control de acceso (ACL).
3. Elija la ACL que desea modificar y elija Modificar
4. En la página Modificar, en Usuarios seleccionados, realice una de las siguientes acciones:
   a. Cree un nuevo usuario seleccionando Crear usuario para agregarlo a la ACL.
   b. Agregue o elimine usuarios seleccionando Administrar y, a continuación, seleccionando o deseleccionando los usuarios en el cuadro de diálogo Administrar usuarios y, a continuación, seleccionando Elegir.
5. En la página Crear lista de control de acceso (ACL), introduzca un nombre ACL.

   Las restricciones para la asignación de nombres de clúster son las siguientes:

   - Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
   - Deben comenzar por una letra.
   - No pueden contener dos guiones consecutivos.
   - No pueden terminar con un guion.
6. En Usuarios seleccionados, realice una de las siguientes acciones:
   a. Para crear un nuevo usuario, seleccione Crear usuario
   b. Para agregar usuarios, elija Administrar y, a continuación, seleccione los usuarios en el cuadro de diálogo Administrar usuarios y, a continuación, seleccione Elegir.
7. Seleccione Modificar para guardar los cambios o Cancelar para descartarlos.

Modificación de una lista de control de acceso (ACL) mediante AWS CLI

Para modificar una ACL agregando usuarios nuevos o eliminando miembros actuales mediante la CLI

- Utilice el comando update-acl para modificar una ACL.

  Para Linux, macOS o Unix:

```
aws memorydb update-acl --acl-name new-acl-1 \
--user-names-to-add user-name-3 \
--user-names-to-remove user-name-2
```

Para Windows:

```
aws memorydb update-acl --acl-name new-acl-1 ^
--user-names-to-add user-name-3 ^
--user-names-to-remove user-name-2
```

> **Note**
>
> Cualquier conexión abierta que pertenezca a un usuario eliminado de una ACL finalizará con este comando.

Información de las listas de control de acceso (ACL) (consola)

Para ver los detalles de la ACL en la consola

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación izquierdo, elija Listas de control de acceso (ACL).
3. Elija la ACL en el nombre de la ACL o utilice el cuadro de búsqueda para buscar la ACL.
4. En Usuarios, puede revisar la lista de usuarios asociados a la ACL.
5. En Clústeres asociados, puede revisar el clúster al que pertenece la ACL.
6. En Etiquetas, puede revisar cualquier etiqueta asociada a la ACL.

Visualización de las listas de control de acceso (ACL) mediante AWS CLI

Utilice el comando describe-acls para ver los detalles de una ACL.

```
aws memorydb describe-acls \
  --acl-name test-group
```

Eliminación de una lista de control de acceso (ACL) (consola)

Para eliminar las listas de control de acceso mediante la consola

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación izquierdo, elija Listas de control de acceso (ACL).
3. Elija la ACL que desee modificar y, a continuación, elija Eliminar
4. En la página de eliminación, ingrese `delete` en el cuadro de confirmación y elija Eliminar o Cancelar para evitar que se elimine la ACL.

Se elimina la ACL en sí, no los usuarios que pertenecen al grupo.

Eliminar una lista de control de acceso (ACL) mediante AWS CLI

Para eliminar una ACL mediante la CLI

- Ejecute el comando delete-acl para eliminar una ACL.

  Para Linux, macOS o Unix:

  ```
  aws memorydb delete-acl /
     --acl-name
  ```

  Para Windows:

  ```
  aws memorydb delete-acl ^
     --acl-name
  ```

  Los ejemplos anteriores devuelven la siguiente respuesta.

  ```
  aws memorydb delete-acl --acl-name "new-acl-1"
  {
      "ACLName": "new-acl-1",
      "Status": "deleting",
      "EngineVersion": "6.2",
      "UserNames": [
          "user-name-1",
          "user-name-3"
      ],
  ```

```
    "clusters": [],
    "ARN":"arn:aws:memorydb:us-east-1:493071037918:acl/new-acl-1"
}
```

Asignación de listas de control de acceso a clústeres

Después de crear una ACL y agregar usuarios, el paso final para implementar las ACL es asignar la ACL a un clúster.

Asignación de listas de control de acceso a los clústeres mediante la consola

Para agregar una ACL a un clúster mediante el AWS Management Console, consulteCreación de un clúster de MemoryDB.

Asignación de listas de control de acceso a clústeres mediante AWS CLI

La siguiente AWS CLI operación crea un clúster con el cifrado en tránsito (TLS) activado y el acl-name parámetro con el valor. *my-acl-name* Reemplace el grupo de subredes `subnet-group` por otro existente.

Parámetros clave

- **`--engine-version`**: debe ser 6.2.
- **`--tls-enabled`**: se utiliza para la autenticación y para asociar una ACL.
- **`--acl-name`**: este valor proporciona listas de control de acceso compuestas por usuarios con permisos de acceso especificados para el clúster.

Para Linux, macOS o Unix:

```
aws memorydb create-cluster \
    --cluster-name "new-cluster" \
    --description "new-cluster" \
    --engine-version "6.2" \
    --node-type db.r6g.large \
    --tls-enabled \
    --acl-name "new-acl-1" \
    --subnet-group-name "subnet-group"
```

Para Windows:

```
aws memorydb create-cluster ^
    --cluster-name "new-cluster" ^
    --cluster-description "new-cluster" ^
    --engine-version "6.2" ^
    --node-type db.r6g.large ^
    --tls-enabled ^
    --acl-name "new-acl-1" ^
    --subnet-group-name "subnet-group"
```

La siguiente AWS CLI operación modifica un clúster con el cifrado en tránsito (TLS) habilitado y el acl-name parámetro con el valor. `new-acl-2`

Para Linux, macOS o Unix:

```
aws memorydb update-cluster \
    --cluster-name cluster-1 \
    --acl-name "new-acl-2"
```

Para Windows:

```
aws memorydb update-cluster ^
    --cluster-name cluster-1 ^
    --acl-name "new-acl-2"
```

Autenticación con IAM

Temas

- Información general
- Limitaciones
- Configuración
- Conexión

Información general

Con la autenticación de IAM, puede autenticar una conexión a MemoryDB mediante identidades de IAM de AWS, cuando el clúster esté configurado para usar la versión 7 o superior de Redis. Esto le permite reforzar el modelo de seguridad y simplificar muchas tareas de seguridad administrativa.

Con la autenticación de IAM puede configurar un control de acceso detallado para cada clúster de MemoryDB y usuario de MemoryDB y seguir los principios de permisos de privilegio mínimo. La autenticación de IAM para MemoryDB para Redis funciona proporcionando un token de autenticación de IAM de corta duración en lugar de una contraseña de usuario de MemoryDB de larga duración en el comando `AUTH` o `HELLO` de Redis. Para obtener más información sobre el token de autenticación de IAM, consulte el proceso de firma de la versión 4 de Signature en la guía de referencia general de AWS y el ejemplo de código que se muestra a continuación.

Puede utilizar las identidades de IAM y sus políticas asociadas para restringir aún más el acceso a Redis. También puede conceder acceso a los usuarios de los proveedores de identidades federados directamente a los clústeres de MemoryDB.

Para usar AWS IAM con MemoryDB para Redis, primero debe crear un usuario de MemoryDB con el modo de autenticación establecido en IAM y, a continuación, puede crear o reutilizar una identidad de IAM. La identidad de IAM necesita una política asociada para conceder la `memorydb:Connect` acción al clúster de MemoryDB y al usuario de MemoryDB. Una vez configurado, puede crear un token de autenticación de IAM con las credenciales de AWS del usuario o rol de IAM. Por último, debe proporcionar el token de autenticación de IAM de corta duración como contraseña en el cliente de Redis cuando se conecte al nodo del clúster de MemoryDB. Un cliente de Redis compatible con el proveedor de credenciales puede generar automáticamente las credenciales temporales para cada nueva conexión. MemoryDB realizará la autenticación de IAM para las solicitudes de conexión de los usuarios de MemoryDB habilitados para IAM y validará las solicitudes de conexión con IAM.

Limitaciones

Si utiliza la autenticación de IAM, se aplicarán las siguientes limitaciones:

- La autenticación de IAM está disponible cuando se utiliza la versión 7.0 o superior del motor de Redis.
- El token de autenticación de IAM es válido durante 15 minutos. Para conexiones de larga duración, recomendamos utilizar un cliente de Redis que admita una interfaz de proveedor de credenciales.
- Una conexión autenticada de IAM a MemoryDB se desconectará automáticamente después de 12 horas. La conexión se puede prolongar durante 12 horas enviando un comando `AUTH` o `HELLO` con un nuevo token de autenticación de IAM.
- Los comandos `MULTI EXEC` no admiten la autenticación de IAM.
- Actualmente, la autenticación de IAM no admite todas las claves de contexto de condición global. Para obtener más información sobre las claves de contexto de condición globales, consulte Claves de contexto de condición globales de AWS en la Guía del usuario de IAM.

Configuración

Para configurar la autenticación de IAM:

1. Crear un clúster

```
aws memorydb create-cluster \
    --cluster-name cluster-01 \
    --description "MemoryDB IAM auth application"
    --node-type db.r6g.large \
    --engine-version 7.0 \
    --acl-name open-access
```

2. Cree un documento de política de confianza de IAM, como se muestra a continuación, para el rol que permita a la cuenta asumir el nuevo rol. Guarde la política en un archivo denominado trust-policy.json.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
        "Action": "sts:AssumeRole"
    }
}
```

3. Cree un documento de política de IAM, como se muestra a continuación. Guarde la política en un archivo denominado policy.json.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect" : "Allow",
      "Action" : [
        "memorydb:connect"
      ],
      "Resource" : [
        "arn:aws:memorydb:us-east-1:123456789012:cluster/cluster-01",
        "arn:aws:memorydb:us-east-1:123456789012:user/iam-user-01"
      ]
    }
```

```
  ]
}
```

4. Cree un rol de IAM.

```
aws iam create-role \
  --role-name "memorydb-iam-auth-app" \
  --assume-role-policy-document file://trust-policy.json
```

5. Cree la política de IAM.

```
aws iam create-policy \
  --policy-name "memorydb-allow-all" \
  --policy-document file://policy.json
```

6. Adjunte la política de IAM al rol.

```
aws iam attach-role-policy \
 --role-name "memorydb-iam-auth-app" \
 --policy-arn "arn:aws:iam::123456789012:policy/memorydb-allow-all"
```

7. Cree un nuevo usuario habilitado para IAM.

```
aws memorydb create-user \
  --user-name iam-user-01 \
  --authentication-mode Type=iam \
  --access-string "on ~* +@all"
```

8. Cree una ACL y asocie al usuario.

```
aws memorydb create-acl \
  --acl-name iam-acl-01 \
  --user-names iam-user-01

aws memorydb update-cluster \
  --cluster-name cluster-01 \
  --acl-name iam-acl-01
```

Conexión

Conectar con el token como contraseña

Primero debe generar el token de autenticación de IAM de corta duración mediante una solicitud prefirmada SigV4 de AWS. Después de eso, debe proporcionar el token de autenticación de IAM como contraseña al conectarse a un clúster de MemoryDB, como se muestra en el ejemplo siguiente.

```
String userName = "insert user name"
String clusterName = "insert cluster name"
String region = "insert region"

// Create a default AWS Credentials provider.
// This will look for AWS credentials defined in environment variables or system
 properties.
AWSCredentialsProvider awsCredentialsProvider = new
 DefaultAWSCredentialsProviderChain();

// Create an IAM authentication token request and signed it using the AWS credentials.
// The pre-signed request URL is used as an IAM authentication token for MemoryDB
 Redis.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userName,
 clusterName, region);
String iamAuthToken =
 iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());

// Construct Redis URL with IAM Auth credentials provider
RedisURI redisURI = RedisURI.builder()
    .withHost(host)
    .withPort(port)
    .withSsl(ssl)
    .withAuthentication(userName, iamAuthToken)
    .build();

// Create a new Lettuce Redis client
RedisClusterClient client = RedisClusterClient.create(redisURI);
client.connect();
```

A continuación, se muestra la definición de `IAMAuthTokenRequest`.

```
public class IAMAuthTokenRequest {
    private static final HttpMethodName REQUEST_METHOD = HttpMethodName.GET;
    private static final String REQUEST_PROTOCOL = "http://";
    private static final String PARAM_ACTION = "Action";
    private static final String PARAM_USER = "User";
    private static final String ACTION_NAME = "connect";
    private static final String SERVICE_NAME = "memorydb";
```

```
    private static final long TOKEN_EXPIRY_SECONDS = 900;

    private final String userName;
    private final String clusterName;
    private final String region;

    public IAMAuthTokenRequest(String userName, String clusterName, String region) {
        this.userName = userName;
        this.clusterName = clusterName;
        this.region = region;
    }

    public String toSignedRequestUri(AWSCredentials credentials) throws
 URISyntaxException {
        Request<Void> request = getSignableRequest();
        sign(request, credentials);
        return new URIBuilder(request.getEndpoint())
            .addParameters(toNamedValuePair(request.getParameters()))
            .build()
            .toString()
            .replace(REQUEST_PROTOCOL, "");
    }

    private <T> Request<T> getSignableRequest() {
        Request<T> request  = new DefaultRequest<>(SERVICE_NAME);
        request.setHttpMethod(REQUEST_METHOD);
        request.setEndpoint(getRequestUri());
        request.addParameters(PARAM_ACTION, Collections.singletonList(ACTION_NAME));
        request.addParameters(PARAM_USER, Collections.singletonList(userName));
        return request;
    }

    private URI getRequestUri() {
        return URI.create(String.format("%s%s/", REQUEST_PROTOCOL, clusterName));
    }

    private <T> void sign(SignableRequest<T> request, AWSCredentials credentials) {
        AWS4Signer signer = new AWS4Signer();
        signer.setRegionName(region);
        signer.setServiceName(SERVICE_NAME);

        DateTime dateTime = DateTime.now();
        dateTime = dateTime.plus(Duration.standardSeconds(TOKEN_EXPIRY_SECONDS));
```

```
        signer.presignRequest(request, credentials, dateTime.toDate());
    }

    private static List<NameValuePair> toNamedValuePair(Map<String, List<String>> in) {
        return in.entrySet().stream()
            .map(e -> new BasicNameValuePair(e.getKey(), e.getValue().get(0)))
            .collect(Collectors.toList());
    }
}
```

Conectar con el proveedor de credenciales

El siguiente código muestra cómo autenticarse con MemoryDB mediante el proveedor de credenciales de autenticación de IAM.

```
String userName = "insert user name"
String clusterName = "insert cluster name"
String region = "insert region"

// Create a default AWS Credentials provider.
// This will look for AWS credentials defined in environment variables or system
 properties.
AWSCredentialsProvider awsCredentialsProvider = new
 DefaultAWSCredentialsProviderChain();

// Create an IAM authentication token request. Once this request is signed it can be
 used as an
// IAM authentication token for MemoryDB Redis.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userName,
 clusterName, region);

// Create a Redis credentials provider using IAM credentials.
RedisCredentialsProvider redisCredentialsProvider = new
 RedisIAMAuthCredentialsProvider(
    userName, iamAuthTokenRequest, awsCredentialsProvider);

// Construct Redis URL with IAM Auth credentials provider
RedisURI redisURI = RedisURI.builder()
    .withHost(host)
    .withPort(port)
    .withSsl(ssl)
    .withAuthentication(redisCredentialsProvider)
    .build();
```

```
// Create a new Lettuce Redis cluster client
RedisClusterClient client = RedisClusterClient.create(redisURI);
client.connect();
```

A continuación, se muestra un ejemplo de un cliente del clúster de Lettuce Redis que incluye IAMAuthTokenRequest en un proveedor de credenciales para generar automáticamente credenciales temporales cuando sea necesario.

```
public class RedisIAMAuthCredentialsProvider implements RedisCredentialsProvider {
    private static final long TOKEN_EXPIRY_SECONDS = 900;

    private final AWSCredentialsProvider awsCredentialsProvider;
    private final String userName;
    private final IAMAuthTokenRequest iamAuthTokenRequest;
    private final Supplier<String> iamAuthTokenSupplier;

    public RedisIAMAuthCredentialsProvider(String userName,
        IAMAuthTokenRequest iamAuthTokenRequest,
        AWSCredentialsProvider awsCredentialsProvider) {
        this.userName = userName;
        this.awsCredentialsProvider = awsCredentialsProvider;
        this.iamAuthTokenRequest = iamAuthTokenRequest;
        this.iamAuthTokenSupplier =
 Suppliers.memoizeWithExpiration(this::getIamAuthToken, TOKEN_EXPIRY_SECONDS,
 TimeUnit.SECONDS);
    }

    @Override
    public Mono<RedisCredentials> resolveCredentials() {
        return Mono.just(RedisCredentials.just(userName, iamAuthTokenSupplier.get()));
    }

    private String getIamAuthToken() {
        return
 iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());
    }
```

Administración de identidades y accesos en MemoryDB para Redis

AWS Identity and Access Management (IAM) es una Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. Los administradores de IAM controlan quién puede estar autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos de MemoryDB. La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

Temas

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo que realice en MemoryDB.

Usuario de servicio: si utiliza el servicio de MemoryDB para realizar su trabajo, su administrador le proporciona las credenciales y los permisos que necesita. A medida que utilice más características de MemoryDB para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una característica en MemoryDB, consulte Solución de problemas de administración de identidades y accesos de MemoryDB para Redis.

Administrador de servicio: si está a cargo de los recursos de MemoryDB en su empresa, es probable que tenga acceso completo a MemoryDB. Es responsabilidad suya determinar a qué características y recursos de MemoryDB deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar IAM con MemoryDB, consulte Cómo funciona MemoryDB para Redis con IAM.

Administrador de IAM: si es un administrador de IAM, es posible que quiera conocer más detalles sobre cómo escribir políticas para administrar el acceso a MemoryDB. Para consultar ejemplos de políticas basadas en la identidad de MemoryDB que puede utilizar en IAM, consulte Ejemplos de políticas basadas en identidades de MemoryDB para Redis.

Autenticación con identidades

La autenticación es la forma en que inicias sesión para AWS usar tus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (IAM Identity Center), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte Firmar las solicitudes de la AWS API en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS Single Sign-On y Uso de la autenticación multifactor (MFA) en AWS en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la

contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios empresarial, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS Single Sign-On. Puede crear usuarios y grupos en el Centro de identidades de IAM o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulte ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales permanentes a largo plazo y los roles proporcionan credenciales temporales. Para más información, consulte Cuándo crear un usuario de IAM (en lugar de un rol) en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente una función de IAM en el AWS Management Console cambiando de función. Puede asumir un rol llamando a una operación de AWS API AWS CLI o utilizando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulte Uso de roles de IAM en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles para federación, consulte Creación de un rol para un proveedor de identidades de terceros en la Guía del usuario de IAM. Si utiliza IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS Single Sign-On.
- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común que ese servicio ejecute

aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.

- Sesiones de acceso directo (FAS): cuando utilizas un usuario o un rol de IAM para realizar acciones en ellas AWS, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.
- Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
- Función vinculada al servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puede usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una instancia EC2 y realizan AWS CLI solicitudes a la API. AWS Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia EC2. Para asignar un AWS rol a una instancia EC2 y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia EC2 obtener credenciales temporales. Para más información, consulte Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2 en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte Cuándo crear un rol de IAM (en lugar de un usuario) en la Guía del usuario de IAM.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Para conceder permiso a los usuarios para realizar acciones en los recursos que necesiten, un administrador de IAM puede crear políticas de IAM. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede adjuntar a una identidad, como un usuario, un grupo de usuarios o un rol de IAM. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios que admiten las ACL. AWS WAF Para obtener más información sobre las ACL, consulte Información general de Lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifique el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.

- Políticas de control de servicios (SCP): las SCP son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y gestionar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilita todas las características en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o a todas sus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una. Usuario raíz de la cuenta de AWS Para obtener más información acerca de Organizations y las SCP, consulte Funcionamiento de las SCP en la Guía del usuario de AWS Organizations.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo AWS determinar si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo funciona MemoryDB para Redis con IAM

Antes de utilizar IAM para administrar el acceso a MemoryDB, conozca qué características de IAM se pueden utilizar con MemoryDB.

Características de IAM que puede utilizar con MemoryDB para Redis

Característica de IAM	Compatibilidad de MemoryDB
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí

Característica de IAM	Compatibilidad de MemoryDB
Recursos de políticas	Sí
Claves de condición de política	No
ACL	Sí
ABAC (etiquetas en políticas)	Sí
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	Sí
Roles vinculados al servicio	Sí

Para obtener una visión general de cómo funcionan MemoryDB y otros AWS servicios con la mayoría de las funciones de IAM, consulte los AWS servicios que funcionan con IAM en la Guía del usuario de IAM.

Políticas basadas en identidades para MemoryDB

Compatibilidad con las políticas basadas en identidades	Sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para más información sobre los elementos que puede utilizar en

una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en identidades de MemoryDB

Para ver ejemplos de políticas basadas en identidad de MemoryDB, consulte Ejemplos de políticas basadas en identidades de MemoryDB para Redis.

Políticas basadas en recursos de MemoryDB

Compatibilidad con las políticas basadas en recursos	No

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para más información, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Acciones de política para MemoryDB

Admite acciones de política	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones de MemoryDB, consulte Acciones definidas por MemoryDB para Redis en la Referencia de autorizaciones de servicio.

Las acciones de políticas de MemoryDB utilizan el siguiente prefijo antes de la acción:

```
MemoryDB
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [
      "MemoryDB:action1",
      "MemoryDB:action2"
         ]
```

Puede utilizar caracteres comodín (*) para especificar varias acciones . Por ejemplo, para especificar todas las acciones que comiencen con la palabra `Describe`, incluya la siguiente acción:

```
"Action": "MemoryDB:Describe*"
```

Para ver ejemplos de políticas basadas en identidad de MemoryDB, consulte Ejemplos de políticas basadas en identidades de MemoryDB para Redis.

Recursos de política para MemoryDB

Admite recursos de políticas	Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Para ver una lista de los tipos de recursos de MemoryDB y sus ARN, consulte Recursos definidos por MemoryDB para Redis en la Referencia de autorizaciones de servicio. Para obtener información sobre las acciones con las que puede especificar el ARN de cada recurso, consulte Acciones definidas por MemoryDB para Redis.

Para ver ejemplos de políticas basadas en identidad de MemoryDB, consulte Ejemplos de políticas basadas en identidades de MemoryDB para Redis.

Claves de condición de política para MemoryDB

Admite claves de condición de políticas específicas del servicio	No

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación lógica `AND`. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulte Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para ver ejemplos de políticas basadas en identidad de MemoryDB, consulte Ejemplos de políticas basadas en identidades de MemoryDB para Redis.

Listas de control de acceso (ACL) de MemoryDB

Admite las ACL	Sí

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Control de acceso basado en atributos (ABAC) con MemoryDB

Admite ABAC (etiquetas en las políticas)	Sí

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte ¿Qué es ABAC? en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Uso de credenciales temporales con MemoryDB

Compatible con el uso de credenciales temporales	Sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluida información sobre cuáles Servicios de AWS funcionan con credenciales temporales, consulta Cómo Servicios de AWS funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para más información sobre el cambio de roles, consulte Cambio a un rol (consola) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para más información, consulte Credenciales de seguridad temporales en IAM.

Permisos de entidades principales entre servicios de MemoryDB

Admite Forward access sessions (FAS)	Sí

Cuando utilizas un usuario o un rol de IAM para realizar acciones en AWSél, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte Reenviar sesiones de acceso.

Roles de servicio para MemoryDB

Compatible con roles de servicio	Sí

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Cómo cambiar los permisos de un rol de servicio podría interrumpir la funcionalidad de MemoryDB. Edite los roles de servicio solo cuando MemoryDB proporcione orientación para hacerlo.

Roles vinculados a servicios para MemoryDB

Compatible con roles vinculados al servicio	Sí

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio

aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM. Busque un servicio en la tabla que incluya `Yes` en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Ejemplos de políticas basadas en identidades de MemoryDB para Redis

De forma predeterminada, los usuarios y roles no tienen permiso para crear ni modificar recursos de MemoryDB. Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS la API. Para conceder permiso a los usuarios para realizar acciones en los recursos que necesiten, un administrador de IAM puede crear políticas de IAM. A continuación, el administrador puede añadir las políticas de IAM a roles, y los usuarios pueden asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

A fin de obtener más información sobre las acciones y los tipos de recursos definidos por MemoryDB, incluido el formato de los ARN para cada tipo de recurso, consulte Acciones, recursos y claves de condición de MemoryDB para Redis en la Referencia de autorizaciones de servicio.

Temas

- Prácticas recomendadas sobre las políticas
- Uso de la consola de MemoryDB
- Cómo permitir a los usuarios consultar sus propios permisos

Prácticas recomendadas sobre las políticas

Las políticas basadas en identidades determinan si alguien puede crear, acceder o eliminar los recursos de MemoryDB de la cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de trabajo en la Guía del usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Política de validación de Analizador de acceso de IAM en la Guía de usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte Configuración del acceso a una API protegido por MFA en la Guía de usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Uso de la consola de MemoryDB

Para acceder a la consola de MemoryDB para Redis, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los recursos de MemoryDB que tiene. Cuenta de AWS Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que concedas permisos mínimos de consola a los usuarios que solo realicen llamadas a la API AWS CLI o a la AWS API. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la consola de MemoryDB, adjunte también la MemoryDB `ConsoleAccess` o la política `ReadOnly` AWS gestionada a las entidades. Para más información, consulte Adición de permisos a un usuario en la Guía del usuario de IAM:

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API o. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
```

```
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Solución de problemas de administración de identidades y accesos de MemoryDB para Redis

Utilice la siguiente información para diagnosticar y solucionar los problemas comunes que puedan surgir cuando trabaje con MemoryDB e IAM.

Temas

- No tengo autorización para realizar una acción en MemoryDB
- No estoy autorizado a realizar lo siguiente: PassRole
- Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis recursos de MemoryDB

No tengo autorización para realizar una acción en MemoryDB

Si AWS Management Console le indica que no está autorizado a realizar una acción, debe ponerse en contacto con el administrador para obtener ayuda. Su administrador es la persona que le facilitó su nombre de usuario y contraseña.

En el siguiente ejemplo, el error se produce cuando el usuario `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `MemoryDB:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 MemoryDB:GetWidget on resource: my-example-widget
```

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al recurso *my-example-widget* mediante la acción `MemoryDB:GetWidget`.

No estoy autorizado a realizar lo siguiente: PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, se deben actualizar las políticas a fin de permitirle pasar un rol a MemoryDB.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, se produce un error cuando un usuario de IAM llamado `marymajor` intenta utilizar la consola para realizar una acción en MemoryDB. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador. AWS El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis recursos de MemoryDB

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para más información, consulte lo siguiente:

- Para obtener información acerca de si MemoryDB admite estas características, consulte Cómo funciona MemoryDB para Redis con IAM.
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro de su propiedad en la Cuenta de AWS Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (identidad federada) en la Guía del usuario de IAM.
- Para obtener información sobre la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte Cómo los roles de IAM difieren de las políticas basadas en recursos en la Guía del usuario de IAM.

Control de acceso

Aunque disponga de credenciales válidas para autenticar las solicitudes, si no tiene permisos, no podrá crear recursos de MemoryDB para Redis ni obtener acceso a ellos. Por ejemplo, debe tener permisos para crear un clúster de MemoryDB.

En las secciones siguientes, se describe cómo administrar los permisos de MemoryDB para Redis. Recomendamos que lea primero la información general.

- Información general sobre la administración de los permisos de acceso a los recursos de MemoryDB
- Uso de políticas basadas en la identidad (políticas de IAM) para MemoryDB para Redis

Información general sobre la administración de los permisos de acceso a los recursos de MemoryDB

Cada AWS recurso es propiedad de una AWS cuenta y los permisos para crear un recurso o acceder a él se rigen por las políticas de permisos. Un administrador de cuentas puede asociar políticas de permisos a identidades de IAM (es decir, usuarios, grupos y funciones). Además, MemoryDB para Redis también permite adjuntar políticas de permisos a los recursos.

> **Note**
>
> Un administrador de cuentas (o usuario administrador) es un usuario que tiene privilegios de administrador. Para obtener más información, consulte Prácticas recomendadas de IAM en la Guía del usuario de IAM.

Para dar acceso, agregue permisos a los usuarios, grupos o roles:

- Usuarios y grupos en AWS IAM Identity Center:

  Cree un conjunto de permisos. Siga las instrucciones de Creación de un conjunto de permisos en la Guía del usuario de AWS IAM Identity Center .
- Usuarios administrados en IAM a través de un proveedor de identidades:

  Cree un rol para la federación de identidades. Siga las instrucciones de Creación de un rol para un proveedor de identidades de terceros (federación) en la Guía del usuario de IAM.
- Usuarios de IAM:
  - Cree un rol que el usuario pueda aceptar. Siga las instrucciones descritas en Creación de un rol para un usuario de IAM en la Guía del usuario de IAM.
  - (No recomendado) Adjunte una política directamente a un usuario o agregue un usuario a un grupo de usuarios. Siga las instrucciones descritas en Adición de permisos a un usuario (consola) de la Guía del usuario de IAM.

Temas

- Recursos y operaciones de MemoryDB para Redis
- Titularidad de los recursos
- Administrar el acceso a los recursos

- Uso de políticas basadas en la identidad (políticas de IAM) para MemoryDB para Redis
- Permisos de nivel de recursos
- Uso de roles vinculados a servicios para Amazon MemoryDB para Redis
- Políticas administradas por AWS para MemoryDB para Redis
- Permisos de la API de MemoryDB: referencia de acciones, recursos y condiciones

Recursos y operaciones de MemoryDB para Redis

En MemoryDB para Redis, el recurso principal es un clúster.

Estos recursos tienen nombres de recursos de Amazon (ARN) únicos asociados a ellos, tal y como se muestra a continuación.

> **Note**
>
> Para que los permisos de nivel de recursos sean efectivos, el nombre del recurso en la cadena de ARN debe estar en minúsculas.

Tipo de recurso	Formato de ARN
Usuario	arn:aws:memorydb:*us-east-1:123456789012* :user/user1
Lista de control de acceso (ACL)	arn:aws:memorydb:*us-east-1:123456789012* :acl/myacl
Clúster	arn:aws:memorydb:*us-east-1:123456789012* :cluster/my-cluster
Instantánea	arn:aws:memorydb:*us-east-1:123456789012* :snapshot/my-snapshot
Grupo de parámetros	*arn:aws:memorydb: us-east-1:123456789012:parameter group/* my-parameter-group

Tipo de recurso	Formato de ARN
Subnet group (Grupo de subredes)	arn:aws:memorydb: *us-east-* 1:1234567 89012:subnetgroup/ my-subnet-group

MemoryDB proporciona un conjunto de operaciones para trabajar con recursos de MemoryDB. Para obtener una lista de operaciones disponibles, consulte Acciones de MemoryDB para Redis.

Titularidad de los recursos

El propietario de un recurso es la cuenta que creó el recurso. AWS Es decir, el propietario del recurso es la AWS cuenta de la entidad principal que autentica la solicitud que crea el recurso. Una entidad principal puede ser la cuenta raíz, un usuario de IAM o un rol de IAM. Los siguientes ejemplos ilustran cómo funciona:

- Supongamos que utiliza las credenciales de la cuenta raíz de su AWS cuenta para crear un clúster. En este caso, su AWS cuenta es la propietaria del recurso. En MemoryDB, el recurso es el clúster.
- Supongamos que crea un usuario de IAM en su AWS cuenta y concede permisos para crear un clúster a ese usuario. En este caso, el usuario puede crear un clúster. Sin embargo, su AWS cuenta, a la que pertenece el usuario, es propietaria del recurso del clúster.
- Supongamos que crea un rol de IAM en su AWS cuenta con permisos para crear un clúster. En este caso, cualquiera que pueda asumir el rol puede crear un clúster. Su AWS cuenta, a la que pertenece el rol, es propietaria del recurso del clúster.

Administrar el acceso a los recursos

Una política de permisos describe quién tiene acceso a qué. En la siguiente sección se explican las opciones disponibles para crear políticas de permisos.

> **Note**
>
> En esta sección se explica el uso de IAM en el contexto de MemoryDB para Redis. No se proporciona información detallada sobre el servicio de IAM. Para ver la documentación completa de IAM, consulte ¿Qué es IAM? en la Guía del usuario de IAM. Para obtener

más información acerca de la sintaxis y las descripciones de las políticas del IAM, consulte Referencia de políticas de IAM de AWS en la Guía del usuario de IAM.

Las políticas que se asocian a una identidad de IAM se denominan políticas basadas en identidades (o políticas de IAM). Las políticas que se adjuntan a un recurso se denominan políticas basadas en recursos.

Temas

- Políticas basadas en identidades (políticas de IAM)
- Especificación de elementos de política: acciones, efectos, recursos y entidades principales
- Especificar las condiciones de una política

Políticas basadas en identidades (políticas de IAM)

Puede asociar políticas a identidades de IAM. Por ejemplo, puede hacer lo siguiente:

- Asociar una política de permisos a un usuario o grupo de la cuenta: un administrador de la cuenta puede utilizar una política de permisos asociada a un usuario determinado para concederle permisos. En este caso, los permisos son para que ese usuario cree un recurso de MemoryDB, como un clúster, un grupo de parámetros o un grupo de seguridad.
- Adjuntar una política de permisos a un rol (conceder permisos para cuentas cruzadas): puede adjuntar una política de permisos basada en identidades a un rol de IAM para conceder permisos para cuentas cruzadas. Por ejemplo, el administrador de la cuenta A puede crear un rol para conceder permisos entre cuentas a otra AWS cuenta (por ejemplo, la cuenta B) o a un AWS servicio de la siguiente manera:
  1. El administrador de la AccountA crea un rol de IAM y asocia una política de permisos a dicho rol, que concede permisos sobre los recursos de la AccountA.
  2. El administrador de la AccountA asocia una política de confianza al rol que identifica la AccountB como la entidad principal que puede asumir el rol.
  3. A continuación, el administrador de la cuenta B puede delegar los permisos para asumir el rol en cualquier usuario de la cuenta B. De este modo, los usuarios de la cuenta B pueden crear o acceder a los recursos de la cuenta A. En algunos casos, es posible que desee conceder permisos a un AWS servicio para que asuma el rol. Para respaldar este enfoque, la entidad principal de la política de confianza también puede ser la entidad principal de un servicio de AWS .

Para obtener más información sobre el uso de IAM para delegar permisos, consulte Administración de accesos en la Guía del usuario de IAM.

El siguiente es un ejemplo de política que permite a un usuario realizar la `DescribeClusters` acción en su AWS cuenta. MemoryDB también permite identificar recursos específicos mediante los ARN de recurso para realizar acciones de la API. Este enfoque también se conoce como "permisos a nivel de recursos".

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Sid": "DescribeClusters",
      "Effect": "Allow",
      "Action": [
         "memorydb:DescribeClusters"],
      "Resource": resource-arn
      }
   ]
}
```

Para obtener más información acerca del uso de políticas basadas en identidades con MemoryDB, consulte Uso de políticas basadas en la identidad (políticas de IAM) para MemoryDB para Redis. Para obtener más información sobre usuarios, grupos, roles y permisos, consulte Identidades (usuarios, grupos y roles) en la Guía del usuario de IAM.

Especificación de elementos de política: acciones, efectos, recursos y entidades principales

En cada recurso de MemoryDB para Redis (consulte Recursos y operaciones de MemoryDB para Redis), el servicio define un conjunto de operaciones de la API (consulte Acciones). Para conceder permisos para estas operaciones de API, MemoryDB define un conjunto de acciones que usted puede especificar en una política. Por ejemplo, para el recurso del clúster de MemoryDB, se definen las siguientes acciones: `CreateCluster`, `DeleteCluster` y `DescribeClusters`. Para realizar una operación API pueden ser necesarios permisos para más de una acción.

A continuación se indican los elementos más básicos de la política:

- Recurso: en una política, se usa un nombre de recurso de Amazon (ARN) para identificar el recurso al que se aplica la política. Para obtener más información, consulte Recursos y operaciones de MemoryDB para Redis.

- Acción: utilice palabras clave de acción para identificar las operaciones del recurso que desea permitir o denegar. Por ejemplo, en función del elemento `Effect` especificado, el permiso `memorydb:CreateCluster` permite o deniega al usuario los permisos para realizar la operación `CreateCluster` de MemoryDB para Redis.
- Efecto: especifique el efecto que se producirá cuando el usuario solicite la acción específica; puede ser permitir o denegar. Si no concede acceso de forma explícita (permitir) a un recurso, el acceso se deniega implícitamente. También puede denegar explícitamente el acceso a un recurso. Por ejemplo, esto puede servir para asegurarse de que un usuario no pueda tener acceso al recurso, aunque otra política le conceda acceso.
- Entidad principal: en las políticas basadas en identidades (políticas de IAM), el usuario al que se asocia esta política es la entidad principal implícita. Para las políticas basadas en recursos, debe especificar el usuario, la cuenta, el servicio u otra entidad que desee que reciba permisos (se aplica solo a las políticas basadas en recursos).

Para obtener más información sobre la sintaxis y descripciones de las políticas de IAM, consulte Referencia de la política de IAM de AWS en la Guía del usuario de IAM.

Para ver una tabla con todas las acciones de la API de MemoryDB para Redis, consulte Permisos de la API de MemoryDB: referencia de acciones, recursos y condiciones.

Especificar las condiciones de una política

Al conceder permisos, puede utilizar el lenguaje de la política de IAM para especificar las condiciones en la que se debe aplicar una política. Por ejemplo, es posible que desee que solo se aplique una política después de una fecha específica. Para obtener más información sobre cómo especificar condiciones en un lenguaje de política, consulte Condition en la Guía del usuario de IAM.

Uso de políticas basadas en la identidad (políticas de IAM) para MemoryDB para Redis

Este tema contiene ejemplos de políticas basadas en identidades, donde los administradores de cuentas pueden asociar políticas de permisos a identidades de IAM (es decir, a usuarios, grupos y funciones).

> **Important**
>
> Le recomendamos que lea primero los temas que explican los conceptos básicos y las opciones para administrar el acceso a sus recursos de MemoryDB para Redis. Para obtener más información, consulte Información general sobre la administración de los permisos de acceso a los recursos de MemoryDB.

En las secciones de este tema se explica lo siguiente:

- Permisos necesarios para usar la consola de MemoryDB para Redis
- Políticas (predefinidas) administradas por AWS para MemoryDB para Redis
- Ejemplos de políticas administradas por los clientes

A continuación se muestra un ejemplo de una política de permisos.

```
{
   "Version": "2012-10-17",
   "Statement": [{
       "Sid": "AllowClusterPermissions",
       "Effect": "Allow",
       "Action": [
          "memorydb:CreateCluster",
          "memorydb:DescribeClusters",
          "memorydb:UpdateCluster"],
       "Resource": "*"
       },
       {
         "Sid": "AllowUserToPassRole",
         "Effect": "Allow",
         "Action": [ "iam:PassRole" ],
         "Resource": "arn:aws:iam::123456789012:role/EC2-roles-for-cluster"
       }
```

```
    ]
}
```

La política tiene dos instrucciones:

- La primera declaración concede permisos para las acciones de MemoryDB para Redis (`memorydb:CreateCluster`, `memorydb:DescribeClusters` y `memorydb:UpdateCluster`) en cualquier clúster que sea propiedad de la cuenta.
- La segunda declaración concede permisos para la acción de IAM (`iam:PassRole`) en el nombre de rol de IAM especificado al final del valor `Resource`.

La política no especifica el elemento `Principal`, ya que en una política basada en la identidad no se especifica el elemento principal que obtiene el permiso. Al asociar una política a un usuario, el usuario es la entidad principal implícita. Cuando asocia una política de permisos a un rol de IAM, el elemento principal identificado en la política de confianza de rol obtiene los permisos.

Para ver una tabla con todas las acciones de la API de MemoryDB para Redis y los recursos a los que se aplican, consulte Permisos de la API de MemoryDB: referencia de acciones, recursos y condiciones.

Permisos necesarios para usar la consola de MemoryDB para Redis

La tabla de referencia de los permisos muestra las operaciones de la API de MemoryDB para Redis e indica los permisos necesarios para cada operación. Para obtener más información sobre las operaciones de la API de MemoryDB, consulte Permisos de la API de MemoryDB: referencia de acciones, recursos y condiciones.

Para usar la consola de MemoryDB para Redis, primero debe conceder permisos para realizar acciones adicionales, tal y como se muestra en la política de permisos siguiente.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "MinPermsForMemDBConsole",
        "Effect": "Allow",
        "Action": [
            "memorydb:Describe*",
            "memorydb:List*",
            "ec2:DescribeAvailabilityZones",
            "ec2:DescribeVpcs",
```

```
            "ec2:DescribeAccountAttributes",
            "ec2:DescribeSecurityGroups",
            "cloudwatch:GetMetricStatistics",
            "cloudwatch:DescribeAlarms",
            "s3:ListAllMyBuckets",
            "sns:ListTopics",
            "sns:ListSubscriptions" ],
        "Resource": "*"
        }
    ]
}
```

La consola de MemoryDB necesita estos permisos adicionales por las siguientes razones:

- Los permisos para las acciones de MemoryDB habilitan la consola para mostrar los recursos de MemoryDB de la cuenta.
- La consola necesita permisos para que las acciones de `ec2` consulten a Amazon EC2 a fin de mostrar las zonas de disponibilidad, VPC, grupos de seguridad y atributos de la cuenta.
- Los permisos para `cloudwatch` las acciones permiten a la consola recuperar CloudWatch las métricas y alarmas de Amazon y mostrarlas en la consola.
- Los permisos para las acciones de `sns` permiten a la consola recuperar suscripciones y temas de Amazon Simple Notification Service (Amazon SNS) y mostrarlos en la consola.

Ejemplos de políticas administradas por los clientes

Si no está utilizando una política predeterminada y elige utilizar una política administrada de forma personalizada, asegúrese de una de las dos cosas. Debería tener permisos para llamar a `iam:createServiceLinkedRole` (para obtener más información, consulte Ejemplo 4: Permitir que un usuario llame a la API de IAM CreateServiceLinkedRole). También puede haber creado un rol vinculado a un servicio de MemoryDB.

Combinadas con los permisos mínimos necesarios para usar la consola de MemoryDB para Redis, las políticas de ejemplo de esta sección conceden permisos adicionales. Los ejemplos también son relevantes para los AWS SDK y el AWS CLI. Para obtener más información acerca de los permisos necesarios para usar la consola de MemoryDB, consulte Permisos necesarios para usar la consola de MemoryDB para Redis.

Para obtener instrucciones sobre la configuración de grupos y usuarios de IAM, consulte Creación del primer grupo y usuario administrador de IAM en la Guía del usuario de IAM.

> **Important**
>
> Pruebe siempre sus políticas de IAM antes de utilizarlas en entornos de producción. Algunas acciones de MemoryDB que parecen sencillas pueden requerir otras acciones de apoyo cuando se usa la consola de MemoryDB. Por ejemplo, `memorydb:CreateCluster` concede permisos para crear clústeres de MemoryDB. Sin embargo, para realizar esta operación, la consola de MemoryDB usa varias acciones `Describe` y `List` para rellenar las listas de la consola.

Ejemplos

- Ejemplo 1: Permitir al usuario acceso de solo lectura a los recursos de MemoryDB
- Ejemplo 2: Concesión de permiso a un usuario para realizar tareas comunes de administrador del sistema de MemoryDB
- Ejemplo 3: Permitir a un usuario obtener acceso a todas las acciones de la API de MemoryDB
- Ejemplo 4: Permitir que un usuario llame a la API de IAM CreateServiceLinkedRole

Ejemplo 1: Permitir al usuario acceso de solo lectura a los recursos de MemoryDB

La política siguiente concede permisos para usar acciones de MemoryDB que permiten a un usuario mostrar recursos. Normalmente, este tipo de política de permisos se adjunta a un grupo de administradores.

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Sid": "MemDBUnrestricted",
      "Effect":"Allow",
      "Action": [
          "memorydb:Describe*",
          "memorydb:List*"],
      "Resource":"*"
      }
   ]
}
```

Ejemplo 2: Concesión de permiso a un usuario para realizar tareas comunes de administrador del sistema de MemoryDB

Entre las tareas comunes de administrador del sistema se incluyen la modificación de clústeres, parámetros y grupos de parámetros. También es posible que el administrador del sistema quiera obtener información acerca de los eventos de MemoryDB. La siguiente política concede a un usuario permisos para realizar acciones de MemoryDB para estas tareas comunes de administrador del sistema. Normalmente, este tipo de política de permisos se adjunta al grupo de administradores del sistema.

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Sid": "MDBAllowSpecific",
      "Effect":"Allow",
      "Action":[
          "memorydb:UpdateCluster",
          "memorydb:DescribeClusters",
          "memorydb:DescribeEvents",
          "memorydb:UpdateParameterGroup",
          "memorydb:DescribeParameterGroups",
          "memorydb:DescribeParameters",
          "memorydb:ResetParameterGroup",],
      "Resource":"*"
      }
   ]
}
```

Ejemplo 3: Permitir a un usuario obtener acceso a todas las acciones de la API de MemoryDB

La siguiente política permite a un usuario obtener acceso a todas las acciones de MemoryDB. Recomendamos que conceda este tipo de política de permisos solo a un usuario administrador.

```
{
   "Version": "2012-10-17",
   "Statement":[{
      "Sid": "MDBAllowAll",
      "Effect":"Allow",
      "Action":[
          "memorydb:*" ],
      "Resource":"*"
      }
```

```
    ]
}
```

Ejemplo 4: Permitir que un usuario llame a la API de IAM CreateServiceLinkedRole

La siguiente política permite al usuario llamar a la API `CreateServiceLinkedRole` de IAM. Le recomendamos que conceda este tipo de política de permisos al usuario que invoca las operaciones de MemoryDB mutantes.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"CreateSLRAllows",
      "Effect":"Allow",
      "Action":[
        "iam:CreateServiceLinkedRole"
      ],
      "Resource":"*",
      "Condition":{
        "StringLike":{
          "iam:AWS ServiceName":"memorydb.amazonaws.com"
        }
      }
    }
  ]
}
```

Permisos de nivel de recursos

Puede restringir el alcance de los permisos de un usuario mediante la especificación de recursos en una política de IAM. Muchas acciones de la AWS CLI API admiten un tipo de recurso que varía en función del comportamiento de la acción. Cada instrucción de una política de IAM concede permiso para realizar una acción en un recurso. Cuando la acción no actúa sobre un recurso nombrado, o cuando se concede permiso para realizar la acción sobre todos los recursos, el valor del recurso en la política es un comodín (*). Para muchas acciones de API, puede restringir los recursos que un usuario puede modificar si especifica el nombre de recurso de Amazon (ARN) de un recurso o un patrón de ARN que coincida con varios recursos. Para restringir los permisos por recurso, especifique el recurso por ARN.

Formato ARN de recursos de MemoryDB

> **Note**
>
> Para que los permisos de nivel de recursos sean efectivos, el nombre del recurso en la cadena de ARN debe estar en minúsculas.

- Usuario: arn:aws:memorydb:*us-east-1:123456789012*:user/user1
- ACL: arn:aws:memorydb:*us-east-1:123456789012*:acl/my-acl
- Clúster: arn:aws:memorydb:*us-east-1:123456789012*:cluster/my-cluster
- Instantánea: arn:aws:memorydb:*us-east-1:123456789012*:snapshot/my-snapshot
- *Grupo de parámetros: arn:aws:memorydb: us-east-1:123456789012:parametergroup/* my-parameter-group
- Grupo de subredes — arn:aws:memorydb: *us-east- 1:123456789012:subnetgroup/* my-subnet-group

Ejemplos

- Ejemplo 1: Permitir a un usuario obtener acceso completo a tipos de recursos de MemoryDB específicos
- Ejemplo 2: Denegarle a un usuario el acceso a un clúster.

Ejemplo 1: Permitir a un usuario obtener acceso completo a tipos de recursos de MemoryDB específicos

La siguiente política permite de forma explícita el acceso completo del `account-id` especificado a todos los recursos de tipo grupo de subredes, grupo de seguridad y clúster.

```
{
        "Sid": "Example1",
        "Effect": "Allow",
        "Action": "memorydb:*",
        "Resource": [
             "arn:aws:memorydb:us-east-1:account-id:subnetgroup/*",
             "arn:aws:memorydb:us-east-1:account-id:securitygroup/*",
             "arn:aws:memorydb:us-east-1:account-id:cluster/*"
        ]
}
```

Ejemplo 2: Denegarle a un usuario el acceso a un clúster.

En el siguiente ejemplo se deniega de forma explícita el acceso del `account-id` especificado a un determinado clúster.

```
{
        "Sid": "Example2",
        "Effect": "Deny",
        "Action": "memorydb:*",
        "Resource": [
                "arn:aws:memorydb:us-east-1:account-id:cluster/name"
        ]
}
```

Uso de roles vinculados a servicios para Amazon MemoryDB para Redis

Amazon MemoryDB para Redis utiliza funciones vinculadas a servicios AWS Identity and Access Management (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a un AWS servicio, como Amazon MemoryDB para Redis. Amazon MemoryDB para Redis predefine los roles vinculados a servicios de Amazon MemoryDB para Redis. Incluyen todos los permisos que requiere el servicio para llamar a otros servicios de AWS en nombre de los clústeres.

Un rol vinculado a un servicio simplifica la configuración de Amazon MemoryDB para Redis dado que no tendrá que agregar los permisos necesarios de forma manual. Los roles ya existen en su AWS cuenta, pero están vinculados a los casos de uso de Amazon MemoryDB for Redis y tienen permisos predefinidos. Solo Amazon MemoryDB para Redis puede asumir estos roles y solo estos roles pueden utilizar la política de permisos predefinida. Las funciones se pueden eliminar únicamente después de eliminar primero sus recursos relacionados. De esta forma se protegen los recursos de Amazon MemoryDB para Redis, ya que evita que se puedan eliminar accidentalmente permisos necesarios para obtener acceso a los recursos.

Para obtener información acerca de otros servicios que admiten roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM y busque los servicios que muestran Sí en la columna Rol vinculado a un servicio. Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado a servicios en cuestión.

Contenido

- Permisos de roles vinculados a servicios para Amazon MemoryDB para Redis
- Creación de un rol vinculado a servicios (IAM)

- Creación de un rol vinculado a servicios (consola de IAM)
- Creación de un rol vinculado a servicios (CLI de IAM)
- Creación de un rol vinculado a servicios (API de IAM)

- Edición de la descripción de un rol vinculado a un servicio para Amazon MemoryDB para Redis
  - Edición de la descripción de un rol vinculado a servicios (consola de IAM)
  - Edición de la descripción de un rol vinculado a servicios (CLI de IAM)
  - Edición de la descripción de un rol vinculado a servicios (API de IAM)
- Eliminación de un rol vinculado a un servicio para Amazon MemoryDB para Redis
  - Limpiar un rol vinculado a un servicio
  - Eliminación de un rol vinculado a servicios (consola de IAM)
  - Eliminación de un rol vinculado a servicios (CLI de IAM)
  - Eliminación de un rol vinculado a servicios (API de IAM)

Permisos de roles vinculados a servicios para Amazon MemoryDB para Redis

Amazon MemoryDB para Redis usa el rol vinculado al servicio denominado AWSServiceRoleForMemoryDB: Esta política permite a MemoryDB administrar los AWS recursos en su nombre según sea necesario para administrar sus clústeres.

La política de permisos de funciones AWSServiceRoleForMemoryDB vinculadas al servicio permite a Amazon MemoryDB for Redis realizar las siguientes acciones en los recursos especificados:

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "ec2:CreateTags"
                ],
                "Resource": "arn:aws:ec2:*:*:network-interface/*",
                "Condition": {
                    "StringEquals": {
                        "ec2:CreateAction": "CreateNetworkInterface"
                    },
                    "ForAllValues:StringEquals": {
                        "aws:TagKeys": [
```

```
                        "AmazonMemoryDBManaged"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteNetworkInterface",
                "ec2:ModifyNetworkInterfaceAttribute"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/AmazonMemoryDBManaged": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteNetworkInterface",
                "ec2:ModifyNetworkInterfaceAttribute"
            ],
            "Resource": "arn:aws:ec2:*:*:security-group/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeSubnets",
```

```
                "ec2:DescribeVpcs"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "AWS/MemoryDB"
                }
            }
        }
    ]
}
```

Para obtener más información, consulte AWSpolítica gestionada: MemoryDBServiceRolePolicy.

Para permitir que una entidad de IAM cree roles vinculados a un servicio AWSServiceRoleForMemoryDB

Agregue la siguiente instrucción de política a los permisos para esa entidad de IAM:

```
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceLinkedRole",
        "iam:PutRolePolicy"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/memorydb.amazonaws.com/
AWSServiceRoleForMemoryDB*",
    "Condition": {"StringLike": {"iam:AWS ServiceName": "memorydb.amazonaws.com"}}
}
```

Para permitir que una entidad de IAM elimine funciones vinculadas al servicio AWSServiceRoleForMemoryDB

Agregue la siguiente instrucción de política a los permisos para esa entidad de IAM:

```
{
```

```
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/memorydb.amazonaws.com/
AWSServiceRoleForMemoryDB*",
    "Condition": {"StringLike": {"iam:AWS ServiceName": "memorydb.amazonaws.com"}}
}
```

Como alternativa, puede usar una política AWS administrada para proporcionar acceso total a Amazon MemoryDB for Redis.

Creación de un rol vinculado a servicios (IAM)

Puede crear un rol vinculado a servicios mediante la consola de IAM, la CLI o la API.

Creación de un rol vinculado a servicios (consola de IAM)

Puede utilizar la consola de IAM para crear un rol vinculado a un servicio.

Para crear un rol vinculado a un servicio (consola)

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo de la consola de IAM, elija Roles. A continuación, elija Create new role (Crear nuevo rol).
3. En Select type of trusted entity (Seleccionar el tipo de entidad de confianza), elija AWS Service (Servicio de).
4. En O seleccione un servicio para ver los casos de uso, elija MemoryDB.
5. Elija Siguiente: permisos.
6. En Policy name (Nombre de la política), tenga en cuenta que `MemoryDBServiceRolePolicy` es necesario para este rol. Elija Siguiente:Etiquetas.
7. Tenga en cuenta que las etiquetas no son compatibles con los roles vinculados a servicios. Elija Next: Review.
8. (Opcional) En Descripción del rol, edite la descripción del nuevo rol vinculado al servicio.
9. Revise el rol y, a continuación, seleccione Crear rol.

Creación de un rol vinculado a servicios (CLI de IAM)

Puede utilizar las operaciones de IAM desde el AWS Command Line Interface para crear un rol vinculado a un servicio. Este rol puede incluir la política de confianza y las políticas insertadas que el servicio necesita para asumir el rol.

Para crear un rol vinculado a un servicio (CLI)

Use la operación siguiente:

```
$ aws iam create-service-linked-role --aws-service-name memorydb.amazonaws.com
```

Creación de un rol vinculado a servicios (API de IAM)

Puede utilizar la API de IAM para crear un rol vinculado a servicios. Este rol puede contener la política de confianza y las políticas insertadas que el servicio necesita para asumir el rol.

Para crear un rol vinculado a un servicio (API)

Use la llamada de API de CreateServiceLinkedRole. En la solicitud, especifique el nombre del servicio de `memorydb.amazonaws.com`.

Edición de la descripción de un rol vinculado a un servicio para Amazon MemoryDB para Redis

Amazon MemoryDB para Redis no le permite editar el AWSServiceRoleForMemoryDB rol vinculado al servicio. Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM.

Edición de la descripción de un rol vinculado a servicios (consola de IAM)

Puede utilizar la consola de IAM para editar una descripción de rol vinculado a servicios.

Para editar la descripción de un rol vinculado a un servicio (consola)

1. En el panel de navegación izquierdo de la consola de IAM, elija Roles.
2. Seleccione el nombre del rol que desea modificar.
3. En el extremo derecho de Role description, seleccione Edit.
4. Ingrese una descripción nueva en el cuadro Save (Guardar).

Edición de la descripción de un rol vinculado a servicios (CLI de IAM)

Puede utilizar las operaciones de IAM desde el para editar la descripción de un rol vinculado AWS Command Line Interface a un servicio.

Para cambiar la descripción de un rol vinculado a un servicio (CLI)

1. (Opcional) Para ver la descripción actual de un rol, utilice la operación AWS CLI for IAM. `get-role`

   Example

   ```
   $ aws iam get-role --role-name AWSServiceRoleForMemoryDB
   ```

   Utilice el nombre del rol, no el ARN, para hacer referencia a los roles con las operaciones de la CLI. Por ejemplo, si una función tiene el ARN `arn:aws:iam::123456789012:role/myrole`, debe referirse a él como **`myrole`**.
2. Para actualizar la descripción de un rol vinculado a un servicio, utilice la operación AWS CLI for IAM. `update-role-description`

   Para Linux, macOS o Unix:

   ```
   $ aws iam update-role-description \
       --role-name AWSServiceRoleForMemoryDB \
       --description "new description"
   ```

   Para Windows:

   ```
   $ aws iam update-role-description ^
       --role-name AWSServiceRoleForMemoryDB ^
       --description "new description"
   ```

Edición de la descripción de un rol vinculado a servicios (API de IAM)

Puede utilizar la API de IAM para editar una descripción de rol vinculado a servicios.

Para cambiar la descripción de un rol vinculado a un servicio (API)

1. (Opcional) Para ver la descripción actual de un rol, utilice la operación de la API de IAM GetRole.

Example

```
https://iam.amazonaws.com/
   ?Action=GetRole
   &RoleName=AWSServiceRoleForMemoryDB
   &Version=2010-05-08
   &AUTHPARAMS
```

2. Para actualizar la descripción de un rol, utilice la operación de la API de IAM UpdateRoleDescription.

   Example

```
https://iam.amazonaws.com/
   ?Action=UpdateRoleDescription
   &RoleName=AWSServiceRoleForMemoryDB
   &Version=2010-05-08
   &Description="New description"
```

Eliminación de un rol vinculado a un servicio para Amazon MemoryDB para Redis

Si ya no necesita usar una característica o servicio que requieran un rol vinculado a un servicio, le recomendamos que elimine dicho rol. De esta forma no tiene una entidad no utilizada que no se monitoree ni mantenga de forma activa. Sin embargo, debe limpiar el rol vinculado al servicio antes de eliminarlo.

Amazon MemoryDB para Redis no elimina de forma automática un rol vinculado a un servicio.

Limpiar un rol vinculado a un servicio

Antes de que pueda utilizar IAM para eliminar un rol vinculado a servicios, primero confirme que el rol no tiene recursos (clústeres) asociados a él.

Para comprobar si el rol vinculado a un servicio tiene una sesión activa en la consola de IAM

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo de la consola de IAM, elija Roles. A continuación, elija el nombre (no la casilla de verificación) del AWSServiceRoleForMemoryDB rol.

3. En la página Resumen del rol seleccionado, seleccione la pestaña Asesor de acceso.
4. En la pestaña Asesor de acceso, revise la actividad reciente del rol vinculado a servicios.

Para eliminar los recursos de Amazon MemoryDB for Redis que requieren AWSServiceRoleForMemoryDB (consola)

- Para eliminar un clúster, consulte los siguientes temas:
  - Usando el AWS Management Console
  - Usando la AWS CLI
  - Uso de la API de MemoryDB

Eliminación de un rol vinculado a servicios (consola de IAM)

Puede utilizar la consola de IAM para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (consola)

1. Inicie sesión en la consola de IAM AWS Management Console y ábrala en https://console.aws.amazon.com/iam/.
2. En el panel de navegación izquierdo de la consola de IAM, elija Roles. A continuación, seleccione la casilla junto al nombre del rol que desea eliminar, no el nombre ni la fila.
3. En Role actions (Acciones de rol) en la parte superior de la página, elija Delete role (Eliminar rol).
4. En la página de confirmación, revise los datos del servicio al que se accedió por última vez, que muestran cuándo accedió por última vez a un AWS servicio cada uno de los roles seleccionados. Esto lo ayuda a confirmar si el rol está actualmente activo. Si desea continuar, seleccione Yes, Delete para enviar la solicitud de eliminación del rol vinculado al servicio.
5. Consulte las notificaciones de la consola de IAM para monitorear el progreso de la eliminación del rol vinculado al servicio. Como el proceso de eliminación del rol vinculado al servicio de IAM es asíncrono, dicha tarea puede realizarse correctamente o fallar después de que envía la solicitud de eliminación. Si la tarea no se realiza correctamente, puede seleccionar View details (Ver detalles) o View Resources (Ver recursos) desde las notificaciones para obtener información sobre el motivo por el que no se pudo eliminar el rol.

Eliminación de un rol vinculado a servicios (CLI de IAM)

Puede utilizar las operaciones de IAM desde allí AWS Command Line Interface para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (CLI)

1. Si no conoce el nombre del rol vinculado a servicios que desea eliminar, ingrese el siguiente comando. En este comando se enumeran los roles y los nombres de recursos de Amazon (ARN) de la cuenta.

   ```
   $ aws iam get-role --role-name role-name
   ```

   Utilice el nombre del rol, no el ARN, para hacer referencia a los roles con las operaciones de la CLI. Por ejemplo, si un rol tiene el ARN `arn:aws:iam::123456789012:role/myrole`, debe referirse a él como **`myrole`**.

2. Como los roles vinculados a servicios no se puede eliminar si están en uso o tienen recursos asociados, debe enviar una solicitud de eliminación. Esta solicitud puede denegarse si no se cumplen estas condiciones. Debe apuntar el valor `deletion-task-id` de la respuesta para comprobar el estado de la tarea de eliminación. Ingrese lo siguiente para enviar una solicitud de eliminación de un rol vinculado a servicios.

   ```
   $ aws iam delete-service-linked-role --role-name role-name
   ```

3. Ingrese lo siguiente para verificar el estado de la tarea de eliminación.

   ```
   $ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
   ```

   El estado de la tarea de eliminación puede ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` o `FAILED`. Si ocurre un error durante la eliminación, la llamada devuelve el motivo del error para que pueda resolver el problema.

Eliminación de un rol vinculado a servicios (API de IAM)

Puede utilizar la API de IAM para eliminar un rol vinculado a un servicio.

Para eliminar un rol vinculado a un servicio (API)

1. Para enviar una solicitud de eliminación de un rol vinculado a un servicio, llame a DeleteServiceLinkedRole. En la solicitud, especifique el nombre del rol.

   Como los roles vinculados a servicios no se puede eliminar si están en uso o tienen recursos asociados, debe enviar una solicitud de eliminación. Esta solicitud puede denegarse si no se cumplen estas condiciones. Debe apuntar el valor `DeletionTaskId` de la respuesta para comprobar el estado de la tarea de eliminación.
2. Para comprobar el estado de la tarea de eliminación, realice una llamada a GetServiceLinkedRoleDeletionStatus. En la solicitud, especifique el valor de `DeletionTaskId`.

   El estado de la tarea de eliminación puede ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` o `FAILED`. Si ocurre un error durante la eliminación, la llamada devuelve el motivo del error para que pueda resolver el problema.

Políticas administradas por AWS para MemoryDB para Redis

Para agregar permisos a usuarios, grupos y roles, es más fácil utilizar las políticas administradas de AWS que escribirlas uno mismo. Se necesita tiempo y experiencia para crear políticas de IAM administradas por el cliente que proporcionen a su equipo solo los permisos necesarios. Para comenzar a hacerlo con rapidez, puede utilizar nuestras políticas administradas por AWS. Estas políticas cubren casos de uso comunes y están disponibles en su cuenta de AWS. Para obtener más información sobre las políticas administradas por AWS, consulte Políticas administradas por AWS en la Guía del usuario de IAM.

Los servicios de AWS mantienen y actualizan las políticas administradas por AWS. No puede cambiar los permisos en las políticas administradas por AWS. En ocasiones, los servicios agregan permisos adicionales a una política administrada por AWS para admitir características nuevas. Este tipo de actualización afecta a todas las identidades (usuarios, grupos y roles) donde se asocia la política. Es más probable que los servicios actualicen una política administrada por AWS cuando se lanza una nueva característica o cuando se ponen a disposición nuevas operaciones. Los servicios no quitan permisos de una política administrada por AWS, por lo que las actualizaciones de políticas no deteriorarán los permisos existentes.

Además, AWS admite políticas administradas para funciones de trabajo que abarcan varios servicios. Por ejemplo, la política administrada por ReadOnlyAccessAWS proporciona acceso de solo lectura a

todos los servicios y los recursos de AWS. Cuando un servicio lanza una nueva característica, AWS agrega permisos de solo lectura para las operaciones y los recursos nuevos. Para obtener una lista y descripciones de las políticas de funciones de trabajo, consulte Políticas administradas de AWS para funciones de trabajo en la Guía del usuario de IAM.

AWSpolítica gestionada: MemoryDBServiceRolePolicy

No puede adjuntar la política gestionada MemoryDBServiceRolePolicy a las AWS identidades de su cuenta. Esta política forma parte del rol vinculado al servicio de AWS MemoryDB. Este rol permite al servicio administrar las interfaces de red y los grupos de seguridad de su cuenta.

MemoryDB usa los permisos de esta política para administrar los grupos de seguridad y las interfaces de red de EC2. Esto es necesario para administrar los clústeres de MemoryDB.

Detalles sobre los permisos

Esta política incluye los siguientes permisos.

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "ec2:CreateTags"
                ],
                "Resource": "arn:aws:ec2:*:*:network-interface/*",
                "Condition": {
                    "StringEquals": {
                        "ec2:CreateAction": "CreateNetworkInterface"
                    },
                    "ForAllValues:StringEquals": {
                        "aws:TagKeys": [
                            "AmazonMemoryDBManaged"
                        ]
                    }
                }
```

```
            },
            {
                "Effect": "Allow",
                "Action": [
                    "ec2:CreateNetworkInterface"
                ],
                "Resource": [
                    "arn:aws:ec2:*:*:network-interface/*",
                    "arn:aws:ec2:*:*:subnet/*",
                    "arn:aws:ec2:*:*:security-group/*"
                ]
            },
            {
                "Effect": "Allow",
                "Action": [
                    "ec2:DeleteNetworkInterface",
                    "ec2:ModifyNetworkInterfaceAttribute"
                ],
                "Resource": "arn:aws:ec2:*:*:network-interface/*",
                "Condition": {
                    "StringEquals": {
                        "ec2:ResourceTag/AmazonMemoryDBManaged": "true"
                    }
                }
            },
            {
                "Effect": "Allow",
                "Action": [
                    "ec2:DeleteNetworkInterface",
                    "ec2:ModifyNetworkInterfaceAttribute"
                ],
                "Resource": "arn:aws:ec2:*:*:security-group/*"
            },
            {
                "Effect": "Allow",
                "Action": [
                    "ec2:DescribeSecurityGroups",
                    "ec2:DescribeNetworkInterfaces",
                    "ec2:DescribeAvailabilityZones",
                    "ec2:DescribeSubnets",
                    "ec2:DescribeVpcs"
                ],
                "Resource": "*"
            },
```

```
            {
                "Effect": "Allow",
                "Action": [
                    "cloudwatch:PutMetricData"
                ],
                "Resource": "*",
                "Condition": {
                    "StringEquals": {
                        "cloudwatch:namespace": "AWS/MemoryDB"
                    }
                }
            }
        ]
    }
```

Políticas (predefinidas) administradas por AWS para MemoryDB para Redis

AWS aborda muchos casos de uso comunes proporcionando políticas de IAM independientes creadas y administradas por AWS. Las políticas administradas conceden los permisos necesarios para casos de uso comunes, lo que le evita tener que investigar los permisos que se necesitan. Para más información, consulte Políticas administradas de AWS en la Guía del usuario de IAM.

Las siguientes políticas administradas de AWS, que se pueden asociar a los usuarios de la cuenta, son específicas de MemoryDB:

AmazonMemoryDBReadOnlyAccess

Puede adjuntar la política `AmazonMemoryDBReadOnlyAccess` a las identidades de IAM. Esta política concede permisos administrativos que permiten acceso de solo lectura a todos los recursos de MemoryDB.

AmazonMemoryDBReadOnlyAccess: concede acceso de solo lectura a los recursos de MemoryDB para Redis.

```
{
 "Version": "2012-10-17",
 "Statement": [{
  "Effect": "Allow",
  "Action": [
   "memorydb:Describe*",
   "memorydb:List*"
  ],
```

```
  "Resource": "*"
 }]
}
```

AmazonMemoryDBFullAccess

Puede adjuntar la política `AmazonMemoryDBFullAccess` a las identidades de IAM. Esta política otorga permisos administrativos que brindan acceso completo a todos los recursos de MemoryDB.

AmazonMemoryDBFullAccess: concede acceso completo a los recursos de MemoryDB para Redis.

```
{
 "Version": "2012-10-17",
 "Statement": [{
   "Effect": "Allow",
   "Action": "memorydb:*",
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": "iam:CreateServiceLinkedRole",
   "Resource": "arn:aws:iam::*:role/aws-service-role/memorydb.amazonaws.com/
AWSServiceRoleForMemoryDB",
   "Condition": {
    "StringLike": {
     "iam:AWSServiceName": "memorydb.amazonaws.com"
    }
   }
  }
 ]
}
```

También puede crear sus propias políticas de IAM personalizadas con el fin de conceder permisos para realizar acciones de la API de MemoryDB para Redis. Puede asociar estas políticas personalizadas a los usuarios o grupos de IAM que requieran esos permisos.

MemoryDB se actualiza a las políticas administradas de AWS

Es posible consultar los detalles sobre las actualizaciones de las políticas administradas de AWS para MemoryDB debido a que este servicio comenzó a realizar el seguimiento de estos cambios.

Para obtener alertas automáticas sobre cambios en esta página, suscríbase a la fuente RSS en la página de historial de documentos de MemoryDB.

Cambio	Descripción	Fecha
AmazonMemoryDBFullAccess: agregar una política	MemoryDB agregó nuevos permisos para describir y enumerar los recursos compatibles. Estos permisos son necesarios para que MemoryDB consulte todos los recursos compatibles de una cuenta.	07/10/2021
AmazonMemoryDBReadOnlyAccess: agregar una política	MemoryDB agregó nuevos permisos para describir y enumerar los recursos compatibles. Estos permisos son necesarios para que MemoryDB cree aplicaciones basadas en cuentas mediante consultas a todos los recursos compatibles de una cuenta.	07/10/2021
MemoryDB comenzó a realizar un seguimiento de los cambios	Lanzamiento del servicio	19/8/2021

Permisos de la API de MemoryDB: referencia de acciones, recursos y condiciones

Cuando configure el control de acceso y escriba políticas de permisos para adjuntar a una política de IAM (políticas basadas en identidad o recurso), utilice la siguiente tabla como referencia. En la tabla se muestran las operaciones de la API de MemoryDB para Redis y las acciones correspondientes para las que puede conceder permisos para realizar la acción. Las acciones se especifican en el campo `Action` de la política y el valor de un recurso se especifica en el campo `Resource` de la política. A menos que se indique lo contrario, el recurso es necesario. Algunos campos incluyen recursos obligatorios y opcionales. Cuando no hay ARN de recurso, el recurso de la política es un comodín (*).

> **Note**
>
> Para especificar una acción, use el prefijo `memorydb:` seguido del nombre de operación de la API (por ejemplo, `memorydb:DescribeClusters`).

Registro y monitorización

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de MemoryDB for Redis y sus demás soluciones. AWS AWS proporciona las siguientes herramientas de monitoreo para ver MemoryDB, informar cuando algo está mal y tomar medidas automáticas cuando sea apropiado:

- Amazon CloudWatch monitorea tus AWS recursos y las aplicaciones en las que AWS ejecutas en tiempo real. Puede recopilar métricas y realizar un seguimiento de las métricas, crear paneles personalizados y definir alarmas que le advierten o que toman medidas cuando una métrica determinada alcanza el umbral que se especifique. Por ejemplo, puede CloudWatch hacer un seguimiento del uso de la CPU u otras métricas de sus instancias de Amazon EC2 y lanzar automáticamente nuevas instancias cuando sea necesario. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.
- Amazon CloudWatch Logs le permite supervisar, almacenar y acceder a sus archivos de registro desde instancias de Amazon EC2 y otras fuentes. CloudTrail CloudWatch Los registros pueden monitorear la información de los archivos de registro y notificarle cuando se alcancen ciertos umbrales. También se pueden archivar los datos del registro en un almacenamiento de larga duración. Para obtener más información, consulta la Guía del usuario CloudWatch de Amazon Logs.

- AWS CloudTrailcaptura las llamadas a la API y los eventos relacionados realizados por su AWS cuenta o en su nombre y entrega los archivos de registro a un bucket de Amazon S3 que especifique. Puede identificar qué usuarios y cuentas llamaron AWS, la dirección IP de origen desde la que se realizaron las llamadas y cuándo se produjeron las llamadas. Para más información, consulte la Guía del usuario de AWS CloudTrail.

Supervisión de MemoryDB para Redis con Amazon CloudWatch

Puede monitorizar MemoryDB para Redis CloudWatch, que recopila datos sin procesar y los procesa para convertirlos en métricas legibles prácticamente en tiempo real. Estas estadísticas se mantienen durante 15 meses, de forma que pueda obtener acceso a información histórica y disponer de una mejor perspectiva sobre el desempeño de su aplicación web o servicio. También puede establecer alarmas que vigilen determinados umbrales y enviar notificaciones o realizar acciones cuando se cumplan dichos umbrales. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.

En las siguientes secciones se detallan las métricas y dimensiones de MemoryDB.

Temas

- Métricas de nivel de host
- Métricas de MemoryDB
- ¿Qué métricas debo monitorear?
- Elección de periodos y estadísticas de métricas
- Monitorear las métricas CloudWatch

Métricas de nivel de host

El espacio de nombres de `AWS/MemoryDB` incluye las siguientes métricas de nivel de host para los distintos nodos.

Véase también

- Métricas de MemoryDB

Métrica	Descripción	Unidad
`CPUUtilization`	El porcentaje de uso de la CPU para todo el host. Dado que Redis utiliza un solo subproceso, recomendamos que monitoree la métrica `EngineCPUUtilization` de los nodos con cuatro o más CPU virtuales.	Porcentaje
`FreeableMemory`	La cantidad de memoria libre disponible en el host. Esto se deriva de la RAM, los búferes y que el sistema operativo indica como liberables.	Bytes
`NetworkBytesIn`	El número de bytes que el host ha leído de la red.	Bytes
`NetworkBytesOut`	El número de bytes enviados en todas las interfaces de red por la instancia.	Bytes
`NetworkPacketsIn`	El número de paquetes recibidos en todas las interfaces de red por la instancia. Esta métrica identifica el volumen de tráfico de red entrante en cuanto al número de paquetes de una sola instancia.	Recuento
`NetworkPacketsOut`	El número de paquetes enviados en todas las interfaces de red por la instancia. Esta métrica identifica el volumen de tráfico de red saliente en cuanto al número de paquetes de una sola instancia.	Recuento
`NetworkBandwidthInAllowanceExceeded`	El número de paquetes formados porque el ancho de banda agregado entrante superó el máximo de la instancia.	Recuento
`NetworkConntrackAllowanceExceeded`	El número de paquetes formados porque el seguimiento de conexiones superó el máximo de la instancia y no se pudieron establecer	Recuento

Métrica	Descripción	Unidad
	nuevas conexiones. Esto puede provocar la pérdida de paquetes para el tráfico hacia o desde la instancia.	
`NetworkBandwidthOutAllowanceExceeded`	El número de paquetes formados porque el ancho de banda agregado saliente superó el máximo de la instancia.	Recuento
`NetworkPacketsPerSecondAllowanceExceeded`	El número de paquetes formados porque los paquetes bidireccionales por segundo superaron el máximo de la instancia.	Recuento
`NetworkMaxBytesIn`	La ráfaga máxima de bytes recibidos en cada minuto.	Bytes
`NetworkMaxBytesOut`	La ráfaga máxima de bytes transmitidos en cada minuto.	Bytes
`NetworkMaxPacketsIn`	La ráfaga máxima de paquetes recibidos en cada minuto.	Recuento
`NetworkMaxPacketsOut`	La ráfaga máxima de paquetes transmitidos en cada minuto.	Recuento
`SwapUsage`	La cantidad de espacio de intercambio utilizado en el host.	Bytes

Métricas de MemoryDB

El espacio de nombres `AWS/MemoryDB` incluye las siguientes métricas de Redis.

Con la excepción de `ReplicationLag` y `EngineCPUUtilization`, estas métricas se obtienen del comando info de Redis. Cada métrica se calcula en el nivel de nodo.

Para ver documentación completa del comando info de Redis, consulte http://redis.io/commands/info.

Véase también

- Métricas de nivel de host

Métrica	Descripción	Unidad
`ActiveDefragHits`	El número de reasignaciones de valor por minuto que ha realizado el proceso de desfragmentación activo. Se obtiene de la estadística de `active_defrag_hits` en Redis INFO.	Número
`AuthenticationFailures`	Número total de intentos fallidos para autenticarse en Redis mediante el comando AUTH. Puede encontrar más información sobre los errores de autenticación individuales mediante el comando ACL LOG. Sugerimos configurar una alarma para detectar intentos de acceso sin autorización.	Recuento
`BytesUsedForMemoryDB`	Número total de bytes asignados por MemoryDB para todos los propósitos, incluido los conjuntos de datos, los búferes, etc.	Bytes
	`Dimension: Tier=SSD` para clústeres que utilizan Organización de datos en niveles: número total de bytes utilizados por SSD.	Bytes
	`Dimension: Tier=Memory` para clústeres que utilizan Organización de datos en niveles: número total de bytes utilizados por memoria. Este es el valor de la estadística de `used_memory` en Redis INFO.	Bytes
`BytesReadFromDisk`	Número total de bytes leídos del disco por minuto. Compatible solo con clústeres que utilizan Organización de datos en niveles.	Bytes

Métrica	Descripción	Unidad
`BytesWrittenToDisk`	Número total de bytes escritos en el disco por minuto. Compatible solo con clústeres que utilizan Organización de datos en niveles.	Bytes
`CommandAuthorizationFailures`	Número total de intentos fallidos de los usuarios de ejecutar comandos a los que no tienen permiso para llamar. Puede encontrar más información sobre los errores de autenticación individuales mediante el comando ACL LOG. Sugerimos configurar una alarma para detectar intentos de acceso sin autorización.	Recuento
`CurrConnections`	Número de conexiones de cliente, excluido las conexiones de réplicas de lectura. MemoryDB utiliza de dos a cuatro de las conexiones para monitorear el clúster en cada caso. Se obtiene de la estadística de `connected_clients` en Redis INFO.	Recuento
`CurrItems`	El número de elementos en la caché. Se obtiene de la estadística `keyspace` de Redis sumando las claves del espacio de claves completo.	Recuento
	`Dimension: Tier=Memory` para clústeres que utilizan Organización de datos en niveles. Número de elementos en la memoria.	Recuento
	`Dimension: Tier=SSD` (unidades de estado sólido) para clústeres que utilizan Organización de datos en niveles. Número de elementos en SSD.	Recuento
`DatabaseMemoryUsagePercentage`	El porcentaje de la memoria disponible para el clúster que está en uso. Esto se calcula usando `used_memory/maxmemory` de Redis INFO.	Porcentaje

Métrica	Descripción	Unidad
`DB0AverageTTL`	Expone `avg_ttl` de DBO a partir de la estadística `keyspace` del comando Redis INFO.	Milisegundos

Métrica	Descripción	Unidad
`EngineCPUUtilization`	Proporciona el uso de la CPU del subproceso del motor de Redis. Como Redis utiliza un solo subproceso, puede utilizar esta métrica para analizar la carga del propio proceso de Redis. La métrica `EngineCPUUtilization` brinda una visibilidad más precisa del proceso de Redis. Puede utilizarla junto con la métrica `CPUUtilization` . `CPUUtilization` muestra el uso de la CPU para la instancia de servidor como un conjunto, lo que incluye otro sistema operativo y los procesos de administr ación. En los tipos de nodos grandes que tienen cuatro o más vCPU, utilice la métrica `EngineCPUUtilization` para monitorear y establecer umbrales de escalado. **Note** En un host de MemoryDB existen procesos en segundo plano que monitorean el host para proporcionar una experiencia de base de datos administrada. Estos procesos en segundo plano pueden ocupar una parte importante de la carga de trabajo de la CPU. Esto no es significativo en host más grandes con más de 2 vCPU. Pero puede afectar a hosts más pequeños con 2 vCPU o menos. Si solo supervisa la métrica `EngineCPU Utilization` , no tendrá constanci a de las situaciones en las que el host esté sobrecargado con un alto uso de CPU de Redis y un alto uso de CPU de los procesos de supervisi	Porcentaje

Métrica	Descripción	Unidad
	ón en segundo plano. Por lo tanto, recomendamos monitorear la métrica `CPUUtilization` en los hosts con 2 vCPU o menos.	
`Evictions`	El número de claves que se han desalojado debido al límite `maxmemory` . Se obtiene de la estadística de `evicted_keys` en Redis INFO.	Recuento
`IsPrimary`	Indica si el nodo es el nodo principal de la partición actual. La métrica puede ser 0 (no principal) o 1 (principal).	Recuento
`KeyAuthorizationFailures`	Número total de intentos fallidos de los usuarios de acceder a claves a las que no tienen permiso para acceder. Puede encontrar más información sobre los errores de autenticación individuales mediante el comando ACL LOG. Sugerimos configurar una alarma para detectar intentos de acceso sin autorización.	Recuento
`KeyspaceHits`	El número de búsquedas de claves solo de lectura realizadas correctamente en el diccionario principal. Se obtiene de la estadística de `keyspace_hits` en Redis INFO.	Recuento
`KeyspaceMisses`	El número de búsquedas de claves solo de lectura que no se realizaron correctamente en el diccionario principal. Se obtiene de la estadística de `keyspace_misses` en Redis INFO.	Recuento

Métrica	Descripción	Unidad
KeysTracked	Número de claves a las que Redis realiza un seguimiento como un porcentaje de `tracking-table-max-keys` . El seguimiento de claves se utiliza para ayudar al almacenamiento en caché del lado del cliente y notifica a los clientes cuando se modifican las claves.	Recuento
MaxReplicationThroughput	El rendimiento máximo de replicación observado durante el último ciclo de medición.	Bytes por segundo
MemoryFragmentationRatio	Indica la eficiencia en la asignación de memoria del motor de Redis. Determina dos umbrales supondrán comportamientos diferentes. El valor recomendado es tener fragmentación por encima de 1,0. Esto se calcula a partir `mem_fragmentation_ratio statistic` de Redis INFO.	Número
NewConnections	El número total de conexiones que ha aceptado el servidor durante este periodo. Se obtiene de la estadística de `total_connections_received` en Redis INFO.	Recuento
NumItemsReadFromDisk	El número total de elementos recuperados del disco por minuto. Compatible solo con clústeres que utilizan Organización de datos en niveles.	Recuento
NumItemsWrittenToDisk	El número total de elementos escritos en disco por minuto. Compatible solo con clústeres que utilizan Organización de datos en niveles.	Recuento

Métrica	Descripción	Unidad
`PrimaryLinkHealthStatus`	Este estado tiene dos valores: 0 o 1. El valor 0 indica que los datos del nodo principal de MemoryDB no se encuentran sincronizados con Redis en EC2. El valor 1 indica que los datos están sincronizados.	Booleano
`Reclaimed`	El número total de eventos de vencimiento de clave. Se obtiene de la estadística de `expired_keys` en Redis INFO.	Recuento
`ReplicationBytes`	Para los nodos en una configuración que se replica, `ReplicationBytes` indica el número de bytes que el nodo principal envía a todas las réplicas. Esta métrica es representativa de la carga de escritura del clúster. Se obtiene de la estadística de `master_repl_offset` en Redis INFO.	Bytes
`ReplicationDelayedWriteCommands`	Número de comandos de escritura que se retrasaron debido a la replicación sincrónica. La replicación se puede retrasar debido a varios factores, por ejemplo, la congestión de la red o la superación del rendimiento máximo de replicación.	Recuento
`ReplicationLag`	Esta métrica solo se aplica a un nodo que se ejecuta como una réplica de lectura. Representa lo que tarda la réplica en aplicar los cambios del nodo principal, en segundos.	Segundos

A continuación se muestran agrupaciones de determinados tipos de comandos, que se obtienen de info commandstats: La sección commandstats proporciona estadísticas basadas en el tipo de comando, incluida la cantidad de llamadas.

Para obtener una lista completa de los comandos disponibles, consulte comandos redis en la documentación de Redis.

Métrica	Descripción	Unidad
EvalBasedCmds	El número total de comandos para los comandos basados en eval. Esto se obtiene de la estadística commandstats de Redis. Esto se obtiene de la estadística commandstats de Redis, mediante la suma de eval, evalsha.	Recuento
GeoSpatialBasedCmds	Número total de comandos para comandos basados en condiciones geoespaciales. Esto se obtiene de la estadística commandstats de Redis. Esto se obtiene al sumar todos los tipos de comandos geográficos: geoadd, geodist, geohash, geopos, georadius y georadiusbymember.	Recuento
GetTypeCmds	El número total de comandos de escritura de read-only. Se obtiene de la estadística commandstats de Redis mediante la suma de todos los tipos de comandos de read-only (get, hget, scard, lrange, etc.).	Recuento
HashBasedCmds	El número total de comandos basados en hash. Se obtiene de la estadística commandstats de Redis mediante la suma de todos los comandos que actúan en uno o más algoritmos hash (hget, hkeys, hvals, hdel, etc.).	Recuento
HyperLogLogBasedCmds	El número total de comandos basados en HyperLogLog . Se obtiene de la estadística commandstats de Redis mediante la suma de todos los tipos de comandos de pf (pfadd, pfcount, pfmerge, etc.).	Recuento

Métrica	Descripción	Unidad
JsonBasedCmds	El número total de comandos basados en JSON. Se obtiene de la estadística commandstats de Redis sumando todos los tipos de comandos que actúan en uno o varios objetos de documento JSON.	Recuento
KeyBasedCmds	El número total de comandos basados en claves. Se obtiene de la estadística commandstats de Redis mediante la suma de todos los comandos que actúan en una o más claves en varias estructuras de datos (del, expire, rename, etc.).	Recuento
ListBasedCmds	El número total de comandos basados en listas. Se obtiene de la estadística commandstats de Redis mediante la suma de todos los comandos que actúan en una o más listas (lindex, lrange, lpush, ltrim, etc.).	Recuento
PubSubBasedCmds	El número total de comandos para la funcionalidad publicación/suscripción. Se obtiene de las estadísticas commandstats de Redis mediante la suma de todos los comandos utilizados para la funcionalidad publicación/suscripción: psubscribe, publish, pubsub, punsubscribe, subscribe y unsubscribe.	Recuento
SearchBasedCmds	El número total de comandos de búsqueda y de índice secundarios, incluidos los comandos de lectura y escritura. Se obtiene a partir de la estadística commandstats de Redis mediante la suma de todos los comandos de búsqueda que actúan sobre índices secundarios.	Recuento

Métrica	Descripción	Unidad
SearchBasedGetCmds	Número total de comandos de solo lectura de índices y búsquedas secundarios. Se obtiene a partir de la estadística commandstats de Redis mediante la suma de todos los comandos de obtener búsqueda e índice secundarios.	Recuento
SearchBasedSetCmds	Número total de comandos de escritura de índices y búsquedas secundarios. Se obtiene a partir de la estadística commandstats de Redis mediante la suma de todos los comandos de configurar búsqueda e índice secundarios.	Recuento
SearchNumberOfIndexes	Número total de índices.	Recuento
SearchNumberOfIndexedKeys	Número total de claves de Redis indexadas	Recuento
SearchTotalIndexSize	Memoria (bytes) utilizada por todos los índices.	Bytes
SetBasedCmds	El número total de comandos basados en instrucciones set. Se obtiene de la estadística commandstats de Redis mediante la suma de todos los comandos que actúan en uno o más conjuntos (scard, sdiff, sadd, sunion, etc.).	Recuento
SetTypeCmds	El número total de tipos de comandos de write. Se obtiene de la estadística commandstats de Redis mediante la suma de todos los tipos de comandos mutative que actúan en los datos (set, hset. sadd, lpop, etc.).	Recuento

Métrica	Descripción	Unidad
`SortedSetBasedCmds`	El número total de comandos basados en instrucciones set ordenadas. Se obtiene de la estadística `commandstats` de Redis mediante la suma de todos los comandos que actúan en uno o más conjuntos ordenados (zcount, zrange, zrank, zadd, etc.).	Recuento
`StringBasedCmds`	El número total de comandos basados en cadenas. Se obtiene de la estadística `commandstats` de Redis mediante la suma de todos los comandos que actúan en una o más cadenas (strlen, setex, setrange, etc.).	Recuento
`StreamBasedCmds`	El número total de comandos basados en secuencias. Se obtiene de la estadística `commandstats` de Redis mediante la suma de todos los comandos que actúan sobre uno o más tipos de datos de secuencia (xrange, xlen, xadd, xdel, etc.).	Recuento

¿Qué métricas debo monitorear?

Las siguientes CloudWatch métricas ofrecen una buena visión del rendimiento de MemoryDB. En la mayoría de los casos, le recomendamos que configure CloudWatch alarmas para estas métricas, de modo que pueda tomar medidas correctivas antes de que se produzcan problemas de rendimiento.

Métricas que se van a monitorear

- CPUUtilization
- EngineCPUUtilization
- SwapUsage
- Evictions
- CurrConnections
- Memoria
- Network
- Replicación

CPUUtilization

Se trata de una métrica de nivel de host que muestra un valor como un porcentaje. Para obtener más información, consulte Métricas de nivel de host.

En los tipos de nodos pequeños que tienen dos CPU virtuales o menos, utilice la métrica `CPUUtilization` para monitorear la carga de trabajo.

En general, sugerimos que establezca el umbral en el 90 % del ancho de banda de la CPU disponible. Debido a que Redis usa un único subproceso, el valor del umbral real debe calcularse como una fracción de la capacidad total del nodo. Por ejemplo, supongamos que está usando un tipo de nodo con dos núcleos. En este caso, el umbral de CPUUtilization sería de 90/2, es decir, el 45 %. Para encontrar el número de núcleos (vCPU) que tiene su tipo de nodo, consulte Precios de MemoryDB.

Deberá determinar su propio umbral en función del número de núcleos del nodo que use. Si supera este umbral y su carga de trabajo principal es de solicitudes de lectura, escale el clúster de forma ascendente agregando réplicas de lectura. Si la carga de trabajo principal es de solicitudes de escritura, recomendamos que agregue más particiones para distribuir la carga de trabajo de escritura entre más nodos principales.

> **Tip**
>
> En lugar de utilizar la métrica de nivel de host `CPUUtilization`, puede utilizar la métrica de Redis `EngineCPUUtilization`, que indica el porcentaje de uso del núcleo del motor de Redis. Para ver si esta métrica está disponible en sus nodos y para obtener más información, consulte Métricas de MemoryDB.

Es posible que, en los tipos de nodos con cuatro o más CPU virtuales, desee utilizar la métrica `EngineCPUUtilization`, que indica el porcentaje de uso del núcleo del motor de Redis. Para ver si esta métrica está disponible en sus nodos y para obtener más información, consulte Métricas de MemoryDB.

EngineCPUUtilization

Es posible que, en los tipos de nodos con cuatro o más CPU virtuales, desee utilizar la métrica `EngineCPUUtilization`, que indica el porcentaje de uso del núcleo del motor de Redis. Para ver si esta métrica está disponible en sus nodos y para obtener más información, consulte Métricas de MemoryDB.

SwapUsage

Se trata de una métrica de nivel de host que muestra un valor en bytes. Para obtener más información, consulte Métricas de nivel de host.

esta métrica no debe superar los 50 MB.

Evictions

Es una métrica del motor. Recomendamos que determine su propio umbral de alarma para esta métrica en función de las necesidades de su aplicación.

CurrConnections

Es una métrica del motor. Recomendamos que determine su propio umbral de alarma para esta métrica en función de las necesidades de su aplicación.

Un número cada vez mayor CurrConnectionspodría indicar un problema con la aplicación; tendrá que investigar el comportamiento de la aplicación para solucionar este problema.

Memoria

La memoria es un aspecto central de Redis. Es necesario comprender la utilización de la memoria de un clúster para evitar la pérdida de datos y adaptarse al crecimiento futuro del conjunto de datos. Las estadísticas sobre la utilización de memoria de un nodo se encuentran disponibles en la sección de memoria sobre el comando INFO de Redis.

Network

Uno de los factores determinantes de la capacidad de la banda ancha de red del clúster es el tipo de nodo seleccionado. Para obtener más información sobre la capacidad de red del nodo, consulte Precios de Amazon MemoryDB.

Replicación

El volumen de datos que se replican es visible a través de la métrica `ReplicationBytes`. Puede realizar un seguimiento del rendimiento de la capacidad de replicación de `MaxReplicationThroughput`. Se recomienda agregar más particiones cuando se alcance el rendimiento máximo de la capacidad de replicación.

`ReplicationDelayedWriteCommands` también puede indicar si la carga de trabajo supera el rendimiento máximo de la capacidad de replicación. Para obtener más información sobre cómo replicar en MemoryDB, consulte Descripción de cómo replicar en MemoryDB

Elección de periodos y estadísticas de métricas

Si bien le CloudWatch permitirá elegir cualquier estadística y período para cada métrica, no todas las combinaciones serán útiles. Por ejemplo, las estadísticas Average, Minimum y Maximum son útiles para CPUUtilization; sin embargo, la estadística Sum no lo es.

Todas las muestras de MemoryDB se publican por un periodo de 60 segundos para cada nodo individual. La métrica de nodo solo contendrá una única muestra para cualquier periodo de 60 segundos.

Monitorear las métricas CloudWatch

MemoryDB y MemoryDB CloudWatch están integrados para que puedas recopilar una variedad de métricas. Puede monitorear estas métricas usando. CloudWatch

> **Note**
>
> Los siguientes ejemplos requieren las herramientas de línea de CloudWatch comandos. Para obtener más información sobre las herramientas para desarrolladores CloudWatch y descargarlas, consulte la página CloudWatch del producto.

Los siguientes procedimientos muestran cómo recopilar las estadísticas de espacio de almacenamiento de un clúster durante la última hora. CloudWatch

> **Note**
>
> Los valores de `StartTime` y `EndTime` proporcionados en los ejemplos siguientes se proporcionan con fines ilustrativos. Debe sustituir los valores de hora de inicio y finalización para sus nodos.

Para obtener información sobre los límites de MemoryDB, consulte los límites de servicio de AWS para MemoryDB.

CloudWatch Métricas de supervisión (consola)

Para recopilar estadísticas de uso de la CPU de un clúster

1. Inicie sesión en la consola de MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. Seleccione los nodos de los que desea ver métricas.

   > **Note**
   >
   > La selección de más de 20 nodos deshabilita la visualización de métricas en la consola.

   a. En la página Clústeres de la consola de AWS administración, haga clic en el nombre de uno o más clústeres.

      Aparecerá la página de detalles del clúster.

   b. Haga clic en la pestaña Nodos situada en la parte superior de la ventana.

   c. En la pestaña Nodos de la ventana de detalles, seleccione los nodos para los que desea ver métricas.

      Aparece una lista de CloudWatch las métricas disponibles en la parte inferior de la ventana de la consola.

   d. Haga clic en la métrica CPU Utilization.

      Se abrirá la CloudWatch consola y mostrará las métricas seleccionadas. Puede usar los cuadros de lista desplegables Statistic y Period y la pestaña Time Range para cambiar las métricas mostradas.

Monitorización CloudWatch de métricas mediante la CloudWatch CLI

Para recopilar estadísticas de uso de la CPU de un clúster

- Utilice el CloudWatch comando aws cloudwatch get-metric-statistics con los siguientes parámetros (tenga en cuenta que las horas de inicio y finalización se muestran solo a modo de ejemplo; tendrá que sustituirlas por las horas de inicio y finalización correspondientes):

  Para Linux, macOS o Unix:

```
aws cloudwatch get-metric-statistics CPUUtilization \
    --dimensions=ClusterName=mycluster,NodeId=0002" \
    --statistics=Average \
```

```
    --namespace="AWS/MemoryDB" \
    --start-time 2013-07-05T00:00:00 \
    --end-time 2013-07-06T00:00:00 \
    --period=60
```

Para Windows:

```
mon-get-stats CPUUtilization ^
    --dimensions=ClusterName=mycluster,NodeId=0002" ^
    --statistics=Average ^
    --namespace="AWS/MemoryDB" ^
    --start-time 2013-07-05T00:00:00 ^
    --end-time 2013-07-06T00:00:00 ^
    --period=60
```

CloudWatch Monitorear las métricas mediante la CloudWatch API

Para recopilar estadísticas de uso de la CPU de un clúster

- Llama a la CloudWatch API `GetMetricStatistics` con los siguientes parámetros (ten en cuenta que las horas de inicio y finalización se muestran solo a modo de ejemplo; tendrás que sustituirlas por las horas de inicio y finalización que correspondan):
  - `Statistics.member.1=Average`
  - `Namespace=AWS/MemoryDB`
  - `StartTime=2013-07-05T00:00:00`
  - `EndTime=2013-07-06T00:00:00`
  - `Period=60`
  - `MeasureName=CPUUtilization`
  - `Dimensions=ClusterName=mycluster,NodeId=0002`

  Example

```
http://monitoring.amazonaws.com/
    ?SignatureVersion=4
    &Action=GetMetricStatistics
    &Version=2014-12-01
```

```
    &StartTime=2013-07-16T00:00:00
    &EndTime=2013-07-16T00:02:00
    &Period=60
    &Statistics.member.1=Average
    &Dimensions.member.1="ClusterName=mycluster"
    &Dimensions.member.2="NodeId=0002"
    &Namespace=Amazon/memorydb
    &MeasureName=CPUUtilization
    &Timestamp=2013-07-07T17%3A48%3A21.746Z
    &AWS;AccessKeyId=<&AWS; Access Key ID>
    &Signature=<Signature>
```

Monitoreo de eventos de MemoryDB para Redis

Cuando se producen eventos significativos en un clúster, MemoryDB envía una notificación a un tema de Amazon SNS concreto. Por ejemplo, errores al agregar un nodo, adiciones de nodos correctas, cambios en un grupo de seguridad, etc. Al monitorear los eventos clave, podrá conocer el estado actual de los clústeres y, dependiendo del evento, adoptar medidas correctivas.

Temas

- Administración de notificaciones de Amazon SNS de MemoryDB
- Visualización de eventos de MemoryDB
- Notificaciones de eventos y Amazon SNS

Administración de notificaciones de Amazon SNS de MemoryDB

Puede configurar MemoryDB para enviar notificaciones de los eventos de clúster importantes mediante Amazon Simple Notification Service (Amazon SNS). En estos ejemplos, podrá configurar un clúster con el nombre de recurso de Amazon (ARN) de un tema de Amazon SNS para recibir notificaciones.

> **Note**
>
> En este tema, se da por sentado que se registró en Amazon SNS, que configuró un tema de Amazon SNS y se suscribió a dicho tema. Para obtener más información sobre cómo realizar esto, consulte la Guía para desarrolladores de Amazon Simple Notification

Adición de un tema de Amazon SNS

En las siguientes secciones, se muestra cómo añadir un tema de Amazon SNS mediante la AWS consola, la o la API AWS CLI de MemoryDB.

Adición de un tema de Amazon SNS (Consola)

En el siguiente procedimiento se muestra cómo agregar un tema de Amazon SNS para un clúster.

> **Note**
>
> Este proceso también se puede utilizar para modificar el tema de Amazon SNS.

A fin de agregar o modificar un tema de Amazon SNS para un clúster (Consola)

1. Inicie sesión en la consola de MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. En Clusters (Clústeres), elija el clúster en el que desee agregar o modificar un ARN de tema de Amazon SNS.
3. Elija Modificar.
4. En Modify Cluster (Modificar clúster) en Topic for SNS Notification (Tema para notificación SNS), elija el tema de SNS que desea agregar, o bien elija Manual ARN input (Entrada manual de ARN) y escriba el ARN del tema de Amazon SNS.
5. Elija Modificar.

Añadir un tema de Amazon SNS (CLI)AWS

Para añadir o modificar un tema de Amazon SNS para un clúster, utilice el AWS CLI comando. `update-cluster`

El siguiente ejemplo de código agrega un ARN de tema de Amazon SNS a my-cluster.

Para Linux, macOS o Unix:

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --sns-topic-arn arn:aws:sns:us-east-1:565419523791:memorydbNotifications
```

Para Windows:

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
    --sns-topic-arn arn:aws:sns:us-east-1:565419523791:memorydbNotifications
```

Para obtener más información, consulte UpdateCluster.

Adición de un tema de Amazon SNS (API de MemoryDB)

Para agregar o actualizar un tema de Amazon SNS para un clúster, realice una llamada a la acción `UpdateCluster` con los siguientes parámetros:

- `ClusterName=my-cluster`
- `SnsTopicArn=arn%3Aaws%3Asns%3Aus-east-1%3A565419523791%3AmemorydbNotifications`

A fin de agregar o actualizar un tema de Amazon SNS para un clúster, realice una llamada a la `UpdateCluster` acción.

Para obtener más información, consulte UpdateCluster.

Habilitación y deshabilitación de notificaciones de Amazon SNS

Puede habilitar o deshabilitar las notificaciones para un clúster. Los siguientes procedimientos muestran cómo deshabilitar las notificaciones de Amazon SNS.

Habilitación y deshabilitación de las notificaciones de Amazon SNS (Consola)

Para deshabilitar las notificaciones de Amazon SNS mediante el AWS Management Console

1. Inicie sesión en la consola MemoryDB for Redis AWS Management Console y ábrala en https://console.aws.amazon.com/memorydb/.
2. Seleccione el botón de opción situado a la izquierda del clúster cuya notificación desea modificar.
3. Elija Modificar.
4. En Modify Cluster, en Topic for SNS Notification, elija Disable Notifications.
5. Elija Modificar.

Activación y desactivación de las notificaciones de Amazon SNS (CLI)AWS

Para deshabilitar las notificaciones de Amazon SNS, utilice el comando `update-cluster` con los siguientes parámetros:

Para Linux, macOS o Unix:

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --sns-topic-status inactive
```

Para Windows:

```
aws memorydb update-cluster ^
    --cluster-name my-cluster ^
    --sns-topic-status inactive
```

Habilitación y deshabilitación de las notificaciones de Amazon SNS (API de MemoryDB)

Para deshabilitar las notificaciones de Amazon SNS, realice una llamada a la acción `UpdateCluster` con los siguientes parámetros:

- `ClusterName=my-cluster`
- `SnsTopicStatus=inactive`

Esta llamada devuelve un resultado similar al siguiente:

Example

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=UpdateCluster
    &ClusterName=my-cluster
    &SnsTopicStatus=inactive
    &Version=2021-01-01
    &SignatureVersion=4
    &SignatureMethod=HmacSHA256
    &Timestamp=20210801T220302Z
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Date=20210801T220302Z
    &X-Amz-SignedHeaders=Host
    &X-Amz-Expires=20210801T220302Z
```

```
    &X-Amz-Credential=<credential>
    &X-Amz-Signature=<signature>
```

Visualización de eventos de MemoryDB

MemoryDB registra eventos relacionados con sus clústeres, grupos de seguridad y grupos de parámetros. Esta información incluye la fecha y la hora del evento, el nombre del origen y el tipo del origen del evento, así como una descripción del evento. Puede recuperar fácilmente los eventos del registro mediante la consola de MemoryDB, el AWS CLI `describe-events` comando o la acción de la API de MemoryDB. `DescribeEvents`

Los procedimientos siguientes muestran cómo ver todos los eventos de MemoryDB de las últimas 24 horas (1440 minutos).

Visualización de eventos de MemoryDB (Consola)

El procedimiento siguiente muestra eventos mediante la consola de MemoryDB.

Para ver eventos mediante la consola de MemoryDB

1. Inicie sesión en la consola MemoryDB AWS Management Console for Redis y ábrala en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación izquierdo, elija Events.

   Aparecerá la pantalla Eventos con todos los eventos disponibles. Cada fila de la lista representa un evento y muestra el origen del evento, el tipo de evento (como cluster, parameter-group, acl, security-group o subnet group), la hora GMT del evento y la descripción del evento.

   Con la opción Filter podrá especificar si desea ver todos los eventos o simplemente los eventos de un tipo determinado de la lista de eventos.

Visualización de eventos de MemoryDB (CLI)AWS

Para generar una lista de eventos de MemoryDB mediante el AWS CLI, utilice el comando. `describe-events` Puede usar parámetros opcionales para controlar el tipo de eventos que se muestran en la lista, el marco temporal de los eventos de la lista, el número máximo de eventos que se incluirán en la lista y mucho más.

El código siguiente muestra hasta 40 eventos de clúster.

```
aws memorydb describe-events --source-type cluster --max-results 40
```

El código siguiente muestra todos los eventos de las últimas 24 horas (1440 minutos).

```
aws memorydb describe-events --duration 1440
```

La salida del comando `describe-events` es similar a la siguiente.

```
{
    "Events": [
        {
            "Date": "2021-03-29T22:17:37.781Z",
            "Message": "Added node 0001 in Availability Zone us-east-1a",
            "SourceName": "memorydb01",
            "SourceType": "cluster"
        },
        {
            "Date": "2021-03-29T22:17:37.769Z",
            "Message": "cluster created",
            "SourceName": "memorydb01",
            "SourceType": "cluster"
        }
    ]
}
```

Para obtener más información como, por ejemplo, los parámetros disponibles y los valores de parámetros permitidos, consulte `describe-events`.

Visualización de eventos de MemoryDB (API de MemoryDB)

Para generar una lista de eventos de MemoryDB mediante la API de MemoryDB, use la acción `DescribeEvents`. Puede usar parámetros opcionales para controlar el tipo de eventos que se muestran en la lista, el marco temporal de los eventos de la lista, el número máximo de eventos que se incluirán en la lista y mucho más.

El código siguiente muestra los 40 eventos de clúster más recientes.

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DescribeEvents
   &MaxResults=40
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &SourceType=cluster
   &Timestamp=20210802T192317Z
   &Version=2021-01-01
```

```
   &X-Amz-Credential=<credential>
```

El código siguiente muestra los eventos de clúster de las últimas 24 horas (1440 minutos).

```
https://memory-db.us-east-1.amazonaws.com/
   ?Action=DescribeEvents
   &Duration=1440
   &SignatureVersion=4
   &SignatureMethod=HmacSHA256
   &SourceType=cluster
   &Timestamp=20210802T192317Z
   &Version=2021-01-01
   &X-Amz-Credential=<credential>
```

Las acciones anteriores deberían producir una salida similar a la siguiente.

```
<DescribeEventsResponse xmlns="http://memory-db.us-east-1.amazonaws.com/
doc/2021-01-01/">
    <DescribeEventsResult>
        <Events>
            <Event>
                <Message>cluster created</Message>
                <SourceType>cluster</SourceType>
                <Date>2021-08-02T18:22:18.202Z</Date>
                <SourceName>my-memorydb-primary</SourceName>
            </Event>

 (...output omitted...)

        </Events>
    </DescribeEventsResult>
    <ResponseMetadata>
        <RequestId>e21c81b4-b9cd-11e3-8a16-7978bb24ffdf</RequestId>
    </ResponseMetadata>
</DescribeEventsResponse>
```

Para obtener más información como, por ejemplo, los parámetros disponibles y los valores de parámetros permitidos, consulte `DescribeEvents`.

Notificaciones de eventos y Amazon SNS

MemoryDB puede publicar mensajes con Amazon Simple Notification Service (SNS) cuando se producen eventos significativos en un clúster. Esta característica se puede usar para actualizar las listas de servidor de las máquinas cliente conectadas a puntos de conexión de nodo individuales de un clúster.

> **Note**
>
> Para obtener más información sobre Amazon Simple Notification Service (SNS), incluido la información sobre los precios y enlaces a los documentos de Amazon SNS, consulte la Página de producto de Amazon SNS.

Las notificaciones se publican en un tema específico de Amazon SNS. A continuación se describen los requisitos para las notificaciones:

- Solo se puede configurar un tema para las notificaciones de MemoryDB.
- La cuenta de AWS que tiene la propiedad del tema de Amazon SNS debe ser la misma cuenta que tiene la propiedad del clúster en el que se encuentran habilitadas las notificaciones.

Eventos de MemoryDB

Los siguientes eventos de MemoryDB desencadenan notificaciones de Amazon SNS:

Nombre del evento	Mensaje	Descripción
MemoryDB:AddNodeComplete	`"Modified number of nodes from %d to %d"`	Se ha agregado un nodo al clúster y está listo para su uso.
MemoryDB:AddNodeFailed a causa de un número insuficiente de direcciones IP libres	`"Failed to modify number of nodes from %d to %d due to insufficient free IP addresses"`	No se pudo agregar un nodo porque no hay suficientes direcciones IP disponibles.

Nombre del evento	Mensaje	Descripción
MemoryDB:ClusterParametersChanged	`"Updated parameter group for the cluster"` In case of create, also send `"Updated to use a ParameterGroup %s"`	Se han cambiado uno o varios parámetros del clúster.
MemoryDB:ClusterProvisioningComplete	`"Cluster created."`	El aprovisionamiento de un clúster se ha completado y los nodos del clúster están listos para el uso.
MemoryDB: ClusterProvisioningFailed debido a un estado de la red incompatible	`"Failed to create cluster due to incompatible network state. %s"`	Se ha intentado lanzar un nuevo clúster en una nube privada virtual (VPC) que no existe.
MemoryDB:ClusterRestoreFailed	`"Restore from %s failed for node %s. %s"`	MemoryDB no pudo rellenar el clúster con los datos de la instantánea de Redis. Esto podría deberse a que el archivo de instantánea de Amazon S3 no existe o a permisos incorrectos en dicho archivo. Si describe el clúster, el estado será `restore-failed` . Deberá eliminar el clúster y comenzar de nuevo. Para obtener más información, consulte Inicialización de un nuevo clúster con una instantánea creada externamente.

Nombre del evento	Mensaje	Descripción
MemoryDB:ClusterScalingComplete	`"Succeeded applying modification to node type to %s."`	El escalado vertical del clúster se ha completado correctamente.
MemoryDB:ClusterScalingFailed	`"Failed applying modification to node type to %s."`	Se ha producido un error en la operación de escalado vertical del clúster.
MemoryDB:ClusterSecurityGroupModified	`"Modified security group for cluster."`	Se ha producido uno de los eventos siguientes: • La lista de los grupos de seguridad autorizados para el clúster se ha modificado. • Se han autorizado uno o varios de los nuevos grupos de seguridad de EC2 en alguno de los grupos de seguridad asociados al clúster. • Se han rechazado uno o varios de los nuevos grupos de seguridad de EC2 de alguno de los grupos de seguridad asociados al clúster.

Nombre del evento	Mensaje	Descripción
MemoryDB:NodeReplaceStarted	`"Recovering node %s"`	MemoryDB ha detectado que el host que ejecuta un nodo tiene un rendimiento reducido o no está disponible y ha comenzado el reemplazo del nodo. **Note** La entrada de DNS del nodo de reemplazo no se ha cambiado. En la mayoría de los casos, cuando se produce este evento, no es necesario actualizar la lista de servidores de sus clientes. Sin embargo, es posible que determinadas bibliotecas del cliente dejen de usar el nodo incluso después de que MemoryDB haya reemplazado el nodo. En este caso, la aplicación deberá actualizar la lista de servidores cuando se produzca este evento.

Nombre del evento	Mensaje	Descripción
MemoryDB:NodeReplaceComplete	`"Finished recovery for node %s"`	MemoryDB ha detectado que el host que ejecuta un nodo tiene un rendimiento reducido o no está disponible y ha completado el reemplazo del nodo. **Note** La entrada de DNS del nodo de reemplazo no se ha cambiado. En la mayoría de los casos, cuando se produce este evento, no es necesario actualizar la lista de servidores de sus clientes. Sin embargo, es posible que determinadas bibliotecas del cliente dejen de usar el nodo incluso después de que MemoryDB haya reemplazado el nodo. En este caso, la aplicación deberá actualizar la lista de servidores cuando se produzca este evento.
MemoryDB:CreateClusterComplete	`"Cluster created"`	El clúster se ha creado correctamente.

Nombre del evento	Mensaje	Descripción
MemoryDB:CreateClusterFailed	`"Failed to create cluster due to unsuccessful creation of its node(s)."` y `"Deleting all nodes belonging to this cluster."`	No se creó el clúster.
MemoryDB:DeleteClusterComplete	`"Cluster deleted."`	Se ha completado la eliminación de un clúster y de todos los nodos asociados.
MemoryDB:FailoverComplete	`"Failover to replica node %s completed"`	La conmutación por error a un nodo de réplica se ha realizado correctamente.
MemoryDB:NodeReplacementCanceled	`"The replacement of node %s which was scheduled during the maintenance window from start time: %s, end time: %s has been canceled"`	Un nodo del clúster que cuyo reemplazo estaba programado ya no está programado para el reemplazo.
MemoryDB:NodeReplacementRescheduled	`"The replacement in maintenance window for node %s has been re-scheduled from previous start time: %s, previous end time: %s to new start time: %s, new end time: %s"`	Un nodo de su clúster que estaba programado para el reemplazo se ha vuelto a programar para el reemplazo durante el nuevo periodo descrito en la notificación. Para obtener información acerca de las acciones que puede emprender, consulte Sustitución de nodos.

Nombre del evento	Mensaje	Descripción
MemoryDB:NodeReplacementScheduled	`"The node %s is scheduled for replacement during the maintenance window from start time: %s to end time: %s"`	Un nodo de su clúster se ha programado para el reemplazo durante el periodo que se describe en la notificación. Para obtener información acerca de las acciones que puede emprender, consulte Sustitución de nodos.
MemoryDB:RemoveNodeComplete	`"Removed node %s"`	Un nodo se ha eliminado del clúster.
MemoryDB:SnapshotComplete	`"Snapshot %s succeeded for node %s"`	Una instantánea se ha completado correctamente.
MemoryDB:SnapshotFailed	`"Snapshot %s failed for node %s"`	se ha producido un error en una de las instantáneas. Consulte los eventos del clúster para obtener más detalles acerca de la causa. Si describe la instantánea (consulte DescribeSnapshots), el estado será `failed`.

Registro de llamadas a la API de MemoryDB para Redis con AWS CloudTrail

MemoryDB para Redis se integra con AWS CloudTrail, un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un servicio de AWS en MemoryDB para Redis. CloudTrail obtiene todas las llamadas a la API para MemoryDB para Redis como eventos, incluido las llamadas procedentes de la consola de MemoryDB para Redis y de las llamadas de código a MemoryDB para las operaciones de la API de Redis. Si crea un registro de seguimiento, puede habilitar la entrega continua de eventos de CloudTrail a un bucket de Amazon S3, incluidos los

eventos para MemoryDB para Redis. Si no configura un registro de seguimiento, puede ver los eventos más recientes de la consola de CloudTrail en el Historial de eventos. Mediante la información que recopila CloudTrail, puede determinar la solicitud que se hizo a los planes de MemoryDB para Redis, la dirección IP desde la que se hizo dicha solicitud, quién la hizo y cuándo, además de información adicional.

Para obtener más información acerca de CloudTrail, consulte la Guía del usuario de AWS CloudTrail.

Información de MemoryDB para Redis en CloudTrail

CloudTrail se habilita en su cuenta de AWS cuando la crea. Cuando se produce una actividad en MemoryDB para Redis, dicha actividad se registra en un evento de CloudTrail junto con los eventos de otros servicios de AWS en Historial de eventos. Puede ver, buscar y descargar los últimos eventos de la cuenta de AWS. Para obtener más información, consulte Ver eventos con el historial de eventos de CloudTrail.

Para mantener un registro continuo de los eventos de su cuenta de AWS, incluidos los eventos de MemoryDB para Redis, cree un registro de seguimiento. Un registro de seguimiento permite a CloudTrail enviar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de Amazon S3 especificado. También es posible configurar otros servicios de AWS para analizar en profundidad y actuar en función de los datos de eventos recopilados en los registros de CloudTrail. Para obtener más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- Servicios e integraciones compatibles con CloudTrail
- Configuración de notificaciones de Amazon SNS para CloudTrail
- Recibir archivos de registro de CloudTrail de varias regiones y Recibir archivos de registro de CloudTrail de varias cuentas

CloudTrail registra todas las acciones de MemoryDB para Redis. Por ejemplo, las llamadas a las acciones `CreateCluster`, `DescribeClusters` y `UpdateCluster` generan entradas en los archivos de registros de CloudTrail.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales raíz o del usuario de IAM de .
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro servicio de AWS.

Para obtener más información, consulte el Elemento userIdentity de CloudTrail.

Descripción de las entradas del archivo de registro de MemoryDB para Redis

Un registro de seguimiento es una configuración que permite la entrega de eventos como archivos de registros en un bucket de Amazon S3 que especifique. Los archivos log de CloudTrail pueden contener una o varias entradas de log. Un evento representa una única solicitud de cualquier origen e incluye información acerca de la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etcétera. Los archivos de registro de CloudTrail no son un rastro de pila ordenado de las llamadas a las API públicas, por lo que no aparecen en ningún orden específico.

En el ejemplo siguiente, se muestra una entrada de registro de CloudTrail que ilustra la acción `CreateCluster`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EKIAUAXQT3SWDEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/john",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "john"
    },
    "eventTime": "2021-07-10T17:56:46Z",
    "eventSource": "memorydb.amazonaws.com",
    "eventName": "CreateCluster",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.01",
    "userAgent": "aws-cli/2.2.29 Python/3.9.6 Darwin/19.6.0 source/x86_64 prompt/off
 command/memorydb.create-cluster",
    "requestParameters": {
        "clusterName": "memorydb-cluster",
        "nodeType": "db.r6g.large",
        "subnetGroupName": "memorydb-subnet-group",
        "aCLName": "open-access"
```

```
    },
    "responseElements": {
        "cluster": {
            "name": "memorydb-cluster",
            "status": "creating",
            "numberOfShards": 1,
            "availabilityMode": "MultiAZ",
            "clusterEndpoint": {
                "port": 6379
            },
            "nodeType": "db.r6g.large",
            "engineVersion": "6.2",
            "enginePatchVersion": "6.2.6",
            "parameterGroupName": "default.memorydb-redis6",
            "parameterGroupStatus": "in-sync",
            "subnetGroupName": "memorydb-subnet-group",
            "tLSEnabled": true,
            "aRN": "arn:aws:memorydb:us-east-1:123456789012:cluster/memorydb-cluster",
            "snapshotRetentionLimit": 0,
            "maintenanceWindow": "tue:06:30-tue:07:30",
            "snapshotWindow": "09:00-10:00",
            "aCLName": "open-access",
            "dataTiering": "false",
            "autoMinorVersionUpgrade": true
        }
    },
    "requestID": "506fc951-9ae2-42bb-872c-98028dc8ed11",
    "eventID": "2ecf3dc3-c931-4df0-a2b3-be90b596697e",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

En el siguiente ejemplo, se muestra una entrada de registro de CloudTrail que ilustra la acción `DescribeClusters`. Tenga en cuenta que se ha eliminado la sección `responseElements` en todas las llamadas de descripción y lista de MemoryDB para Redis (`Describe*` y `List*`) y ahora se muestra como `null`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
```

```
        "type": "IAMUser",
        "principalId": "EKIAUAXQT3SWDEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/john",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "john"
    },
    "eventTime": "2021-07-10T18:39:51Z",
    "eventSource": "memorydb.amazonaws.com",
    "eventName": "DescribeClusters",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.01",
    "userAgent": "aws-cli/2.2.29 Python/3.9.6 Darwin/19.6.0 source/x86_64 prompt/off
 command/memorydb.describe-clusters",
    "requestParameters": {
        "maxResults": 50,
        "showShardDetails": true
    },
    "responseElements": null,
    "requestID": "5e831993-52bb-494d-9bba-338a117c2389",
    "eventID": "32a3dc0a-31c8-4218-b889-1a6310b7dd50",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

En el ejemplo siguiente se muestra una entrada de log de CloudTrail que registra una acción `UpdateCluster`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EKIAUAXQT3SWDEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/john",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "john"
    },
    "eventTime": "2021-07-10T19:23:20Z",
    "eventSource": "memorydb.amazonaws.com",
```

```
    "eventName": "UpdateCluster",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.01",
    "userAgent": "aws-cli/2.2.29 Python/3.9.6 Darwin/19.6.0 source/x86_64 prompt/off
 command/memorydb.update-cluster",
    "requestParameters": {
        "clusterName": "memorydb-cluster",
        "snapshotWindow": "04:00-05:00",
        "shardConfiguration": {
            "shardCount": 2
        }
    },
    "responseElements": {
        "cluster": {
            "name": "memorydb-cluster",
            "status": "updating",
            "numberOfShards": 2,
            "availabilityMode": "MultiAZ",
            "clusterEndpoint": {
                "address": "clustercfg.memorydb-cluster.cde8da.memorydb.us-
east-1.amazonaws.com",
                "port": 6379
            },
            "nodeType": "db.r6g.large",
            "engineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "parameterGroupName": "default.memorydb-redis6",
            "parameterGroupStatus": "in-sync",
            "subnetGroupName": "memorydb-subnet-group",
            "tLSEnabled": true,
            "aRN": "arn:aws:memorydb:us-east-1:123456789012:cluster/memorydb-cluster",
            "snapshotRetentionLimit": 0,
            "maintenanceWindow": "tue:06:30-tue:07:30",
            "snapshotWindow": "04:00-05:00",
            "autoMinorVersionUpgrade": true,
            "DataTiering": "false"
        }
    },
    "requestID": "dad021ce-d161-4365-8085-574133afab54",
    "eventID": "e0120f85-ab7e-4ad4-ae78-43ba15dee3d8",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
```

```
    "eventCategory": "Management"
}
```

En el siguiente ejemplo, se muestra una entrada de registro de CloudTrail que ilustra la acción `CreateUser`. Tenga en cuenta que, en el caso de las llamadas a MemoryDB para Redis que contengan datos confidenciales, dichos datos se redactarán en el evento de CloudTrail correspondiente, tal y como se muestra en la sección siguiente `requestParameters`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EKIAUAXQT3SWDEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/john",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "john"
    },
    "eventTime": "2021-07-10T19:56:13Z",
    "eventSource": "memorydb.amazonaws.com",
    "eventName": "CreateUser",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.01",
    "userAgent": "aws-cli/2.2.29 Python/3.9.6 Darwin/19.6.0 source/x86_64 prompt/off
 command/memorydb.create-user",
    "requestParameters": {
        "userName": "memorydb-user",
        "authenticationMode": {
            "type": "password",
            "passwords": [
                "HIDDEN_DUE_TO_SECURITY_REASONS"
            ]
        },
        "accessString": "~* &* -@all +@read"
    },
    "responseElements": {
        "user": {
            "name": "memorydb-user",
            "status": "active",
            "accessString": "off ~* &* -@all +@read",
            "aCLNames": [],
            "minimumEngineVersion": "6.2",
            "authentication": {
```

```
                "type": "password",
                "passwordCount": 1
            },
            "aRN": "arn:aws:memorydb:us-east-1:123456789012:user/memorydb-user"
        }
    },
    "requestID": "ae288b5e-80ab-4ff8-989a-5ee5c67cd193",
    "eventID": "ed096e3e-16f1-4a23-866c-0baa6ec769f6",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```


Validación de la conformidad en MemoryDB para Redis

Los auditores externos evalúan la seguridad y la conformidad de MemoryDB para Redis en distintos programas de conformidad de AWS. Esto incluye:

- Estándar de Seguridad de Datos del Sector de las Tarjetas de Pago (PCI DSS, Payment Card Industry Data Security Standard). Para obtener más información, consulte PCI DSS.
- Acuerdo para socio empresarial de la ley de portabilidad y responsabilidad de seguros médicos (HIPAA BAA). Para obtener más información, consulte Conformidad con HIPAA.
- Controles del Sistema y Organizaciones (System and Organization Controls, SOC) 1, 2 y 3. Para obtener más información, consulte SOC.
- Programa Federal de Administración de Riesgos y Autorizaciones (Federal Risk and Authorization Management Program, FedRAMP). Para obtener más información, consulte FedRAMP.
- ISO/IEC 27001:2013, 27017:2015, 27018:2019 e ISO/IEC 9001:2015. Para obtener más información, consulte Certificaciones y servicios ISO y CSA STAR de AWS.

Para obtener una lista de los servicios que AWS incluyen los programas de conformidad específicos, consulte los servicios AWS incluidos en cada programa de conformidad.

Puede descargar los informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact.

Su responsabilidad de conformidad al utilizar MemoryDB se determina en función de la sensibilidad de los datos, los objetivos de cumplimiento de su empresa y la legislación y los reglamentos correspondientes. AWS proporciona los siguientes recursos para ayudar con la conformidad:

- Guías de inicio rápido de seguridad y conformidad: estas guías de implementación tratan consideraciones sobre arquitectura y ofrecen pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- Recursos de conformidad de AWS: este conjunto de manuales y guías podría aplicarse a su sector y ubicación.
- Evaluación de recursos con reglas en la Guía para desarrolladores de AWS Config: AWS Config evalúa en qué medida las configuraciones de sus recursos cumplen las prácticas internas, las directrices del sector y las normativas .
- AWS Security Hub: este servicio de AWS proporciona una vista integral de su estado de seguridad en AWS que lo ayuda a verificar la conformidad con los estándares y las prácticas recomendadas del sector de seguridad.
- AWS Audit Manager: este servicio de AWS le ayuda a auditar de manera continua su uso de AWS para simplificar la forma en que gestiona el riesgo y la conformidad con las regulaciones y los estándares del sector.

Seguridad de la infraestructura en Amazon MemoryDB para Redis

Al tratarse de un servicio administrado, MemoryDB está protegido por los procedimientos de seguridad de red globales de AWS que se describen en el documento técnico Amazon Web Services: Información general sobre los procesos de seguridad.

Puede utilizar llamadas a la API publicada de AWS para obtener acceso a MemoryDB a través de la red. Los clientes deben ser compatibles con Transport Layer Security (TLS) 1.2 o una versión posterior. Recomendamos TLS 1.3 o una versión posterior. Los clientes también deben ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Privacidad del tráfico entre redes

MemoryDB para Redis usa las siguientes técnicas para proteger los datos frente a accesos no autorizados:

- MemoryDB y Amazon VPC explica el tipo de grupo de seguridad que necesita para su instalación.
- Puntos de conexión de la VPC de interfaz y API de MemoryDB para Redis (AWS PrivateLink) permite establecer una conexión privada entre la VPC y MemoryDB para los puntos de conexión de la API de MemoryDB para Redis.
- Administración de identidades y accesos en MemoryDB para Redis para conceder y limitar las acciones de los usuarios, grupos y roles.

MemoryDB y Amazon VPC

El servicio de Amazon Virtual Private Cloud (Amazon VPC) define una red virtual que se parece mucho a un centro de datos tradicional. Al configurar su nube privada virtual (VPC) con Amazon VPC, puede seleccionar su rango de direcciones IP, crear subredes y configurar las tablas de enrutamiento, las puertas de enlace de red y la configuración de seguridad. También puede agregar un clúster a la red virtual y controlar el acceso al clúster mediante los grupos de seguridad de Amazon VPC.

En esta sección se explica cómo configurar manualmente un clúster de MemoryDB en una VPC. Esta información se ha pensado para usuarios que desean conocer en detalle cómo funcionan MemoryDB y Amazon VPC de manera conjunta.

Temas

- Comprensión de MemoryDB y VPC
- Patrones de acceso para obtener acceso a un clúster de MemoryDB en una Amazon VPC
- Creación de una Virtual Private Cloud (VPC)

Comprensión de MemoryDB y VPC

MemoryDB está totalmente integrado con Amazon VPC. Para los usuarios de MemoryDB, esto significa lo siguiente:

- MemoryDB siempre lanza el clúster en una VPC.
- Si es la primera vez que lo haceAWS, se creará automáticamente una VPC predeterminada.
- Si tiene una VPC predeterminada y no especifica una subred al lanzar un clúster, el clúster se lanzará en su Amazon VPC predeterminada.

Para obtener más información, consulte Detección de plataformas compatibles y comprobación de si tiene una VPC predeterminada.

Con Amazon VPC, puede crear una red virtual en la nube de AWS que se parece mucho a un centro de datos tradicional. Puede configurar su VPC, así como seleccionar el rango de direcciones IP, crear subredes o configurar las tablas de enrutamiento, las puertas de enlace de red y la configuración de seguridad.

MemoryDB administra las actualizaciones de software, la detección de errores y la recuperación.

Descripción general de MemoryDB en una VPC

1	Una VPC es una parte aislada de la nube de AWS a la que se asigna su propio bloque de direcciones IP.
2	Una puerta de enlace de Internet conecta la VPC directamente a Internet y proporciona acceso a otros recursos de AWS como Amazon Simple Storage Service (Amazon S3) que se ejecutan fuera de su VPC.
3	Una subred de Amazon VPC es un segmento del rango de direcciones IP de una VPC donde podrá aislar recursos de AWS en función de sus necesidades operativas y de seguridad.
4	Una tabla de enrutamiento en su VPC dirige el tráfico de red entre la subred e Internet. La Amazon VPC tiene un router implícito.

5	Un grupo de seguridad de Amazon VPC controla el tráfico entrante y saliente de sus clústeres de MemoryDB y sus instancias de Amazon EC2.
6	Puede lanzar un clúster de MemoryDB en la subred. Los nodos tienen direcciones IP privadas del rango de direcciones de la subred.
7	También puede lanzar instancias de Amazon EC2 en la subred. Cada instancia de Amazon EC2 tiene una dirección IP privada del rango de direcciones de la subred. La instancia de Amazon EC2 puede conectarse a cualquier nodo de la misma subred.
8	Para que se pueda obtener acceso a una instancia de Amazon EC2 de su VPC desde Internet, deberá asignar a la instancia una dirección pública y estática denominada dirección IP elástica.

Requisitos previos

Para crear un clúster de MemoryDB en una VPC, la VPC debe cumplir los requisitos siguientes:

- Su VPC debe permitir instancias de Amazon EC2 no dedicadas. No puede usar MemoryDB en una VPC configurada para la tenencia de instancias dedicadas.
- Debe definir un grupo de subredes para su VPC. MemoryDB utiliza dicho grupo de subredes para seleccionar una subred y direcciones IP pertenecientes a ella, así como para asociárselas a los nodos.
- Debe definir un grupo de seguridad para su VPC, o bien puede usar el grupo predeterminado facilitado.
- Los bloques de CIDR de cada subred deben ser lo suficientemente grandes como para proporcionar direcciones IP auxiliares de MemoryDB que puedan usarse durante las actividades de mantenimiento.

Enrutamiento y seguridad

Puede configurar el enrutamiento en su VPC; para controlar dónde fluye el tráfico (por ejemplo, a la puerta de enlace de Internet o la puerta de enlace privada virtual). Con una puerta de enlace de Internet, su VPC; tiene acceso directo a otros recursos de AWS que no se están ejecutando

en su VPC. Si decide tener solo una gateway privada virtual con una conexión a la red local de su organización, puede enrutar el tráfico vinculado a Internet a través de la VPN y utilizar políticas de seguridad locales y firewalls para controlar las salidas. En ese caso, se cobrarán tarifas de banda ancha adicionales al obtener acceso a los recursos de AWS a través de Internet.

Puede utilizar grupos de seguridad de Amazon VPC para ayudar a proteger los clústeres de MemoryDB y las instancias de Amazon EC2 de su Amazon VPC. Los grupos de seguridad actúan como un firewall en el ámbito de la instancia, no en el de la subred.

> **Note**
>
> Recomendamos utilizar nombres de DNS para conectarse a los nodos, ya que la dirección IP subyacente puede cambiar con el tiempo.

Documentación de Amazon VPC

Amazon VPC tiene su propia serie de documentación que describe cómo crear y utilizar una Amazon VPC. En la siguiente tabla, se indica dónde encontrar información en las guías de Amazon VPC.

Descripción	Documentación
Cómo comenzar a utilizar Amazon VPC	Introducción a Amazon VPC
Cómo utilizar Amazon VPC a través de la AWS Management Console	Guía del usuario de Amazon VPC
Descripciones completas de todos los comandos de Amazon VPC	Referencia de línea de comandos de Amazon EC2 (los comandos de Amazon VPC se encuentran en la referencia de Amazon EC2)
Descripciones completas de las operaciones, los tipos de datos y los errores de la API de Amazon VPC	Referencia de la API de Amazon EC2 (las operaciones de la API de Amazon VPC se encuentran en la referencia de Amazon EC2)
Información para el administrador de red que necesita configurar la gateway en su extremo para disponer de una conexión de VPN IPsec opcional	¿Qué es AWS Site-to-Site VPN?

Para obtener información más detallada sobre Amazon Virtual Private Cloud, consulte Amazon Virtual Private Cloud.

Patrones de acceso para obtener acceso a un clúster de MemoryDB en una Amazon VPC

MemoryDB para Redis admite los siguientes escenarios para obtener acceso a un clúster en una Amazon VPC:

Contenido

Acceso a un clúster de MemoryDB cuando este y la instancia de Amazon EC2 se encuentran en la misma Amazon VPC

El caso de uso más común es cuando una aplicación implementada en una instancia EC2 debe conectarse a un clúster en la misma VPC.

La forma más sencilla de administrar el acceso entre instancias EC2 y clústeres de la misma VPC es realizar lo siguiente:

1. Cree un grupo de seguridad de VPC para su clúster. Este grupo de seguridad se puede utilizar para restringir el acceso a los clústeres. Por ejemplo, puede crear una regla personalizada para este grupo de seguridad que permita el acceso mediante TCP utilizando el puerto que asignó al

clúster de base de datos cuando lo creó y una dirección IP que se utilizará para obtener acceso al clúster.

El puerto predeterminado para los clústeres de MemoryDB es 6379.

2. Cree un grupo de seguridad de VPC para sus instancias EC2 (servidores web y de aplicaciones). Si es necesario, este grupo de seguridad puede permitir el acceso a la instancia EC2 desde Internet a través de la tabla de enrutamiento de la VPC. Por ejemplo, puede establecer reglas en este grupo de seguridad para permitir el acceso mediante TCP a la instancia EC2 a través del puerto 22.
3. Cree reglas personalizadas en el grupo de seguridad para su clúster que permitan las conexiones desde el grupo de seguridad que creó para las instancias EC2. Esto permitirá a cualquier miembro del grupo de seguridad obtener acceso a los clústeres.

Para crear una regla en un grupo de seguridad de VPC que permita establecer conexiones desde otro grupo de seguridad

1. Inicie sesión en la consola de administración de AWS y abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc.
2. En el panel de navegación izquierdo, elija Security Groups.
3. Seleccione o cree un grupo de seguridad que utilizará para sus clústeres. En Inbound Rules (Reglas de entrada), seleccione Edit Inbound Rules (Editar reglas de entrada) y, a continuación, seleccione Add Rule (Agregar regla). Este grupo de seguridad permitirá el acceso a los miembros de otro grupo de seguridad.
4. En Type (Tipo), elija Custom TCP Rule (Personalizar regla de TCP).
   a. En Port Range (Rango de puerto), especifique el puerto que utilizó al crear su clúster.

      El puerto predeterminado para los clústeres de MemoryDB es 6379.
   b. En el cuadro Source (Fuente), comience a escribir el ID del grupo de seguridad. Desde la lista, seleccione el grupo de seguridad que utilizará para sus instancias de Amazon EC2.
5. Cuando haya terminado, elija Save (Guardar).

Acceso a un clúster de MemoryDB cuando este y la instancia de Amazon EC2 se encuentran en Amazon VPC diferentes

Cuando un clúster se encuentra en una VPC que no coincide con la de la instancia EC2 que se utiliza para acceder a él, existen varias formas de obtener acceso al clúster. Si el clúster y la instancia EC2 están en VPC distintas, pero en la misma región, puede utilizar la conexión de emparejamiento de VPC. Si el clúster y la instancia EC2 se encuentran en distintas regiones, puede crear conectividad de VPN entre regiones.

Temas

Acceso a un clúster de MemoryDB cuando este y la instancia de Amazon EC2 se encuentran en Amazon VPC diferentes en la misma región

Acceso a clúster de una instancia de Amazon EC2 en una Amazon VPC diferente dentro de la misma región, conexión de emparejamiento de VPC

Una conexión de emparejamiento de VPC es una conexión de redes entre dos VPC que permite direccionar el tráfico entre ellas mediante direcciones IP privadas. Las instancias de ambas VPC se pueden comunicar entre sí siempre que se encuentren en la misma red. Puede crear una conexión de emparejamiento de VPC entre sus propias Amazon VPC o con una Amazon VPC de otra cuenta de AWS dentro de una misma región. Para obtener más información sobre la interconexión de Amazon VPC, consulte la documentación de VPC.

Para obtener acceso a un clúster en una Amazon VPC diferente a través de interconexiones

1. Asegúrese de que las dos VPC no tengan rangos de IP solapados o no podrá interconectarlas.
2. Coloque las dos VPC al mismo nivel. Para obtener más información, consulte Creación y aceptación de interconexiones de Amazon VPC.
3. Actualice su tabla de ruteo. Para obtener más información, consulte Actualización de las tablas de ruteo para interconexiones de VPC

4. Modifique el grupo de seguridad de su clúster de MemoryDB para permitir la conexión de entrada del grupo de seguridad de la aplicación en la VPC del mismo nivel. Para obtener más información, consulte Actualización de los grupos de seguridad para que hagan referencia a grupos de la VPC del mismo nivel.

El acceso a un clúster a través de una conexión de emparejamiento generará costos de transferencia de datos adicionales.

Uso de Transit Gateway

Una gateway de tránsito permite adjuntar las VPC y las conexiones de VPN de la misma región de AWS y enrutar el tráfico entre ellas. Una gateway de tránsito funciona en todas las cuentas de AWS y puede utilizar AWS Resource Access Manager para compartir la gateway de tránsito con otras cuentas. Después de compartir una transit gateway con otra cuenta de AWS, el propietario de la cuenta puede asociar las VPC a la gateway de tránsito. Un usuario de cualquiera de las cuentas puede eliminar la vinculación en cualquier momento.

Puede habilitar la multidifusión en una gateway de tránsito y, a continuación, crear un dominio de multidifusión de transit gateway que permita que el trafico de multidifusión se envíe desde el origen de multidifusión a los miembros del grupo de multidifusión a través de conexiones de la VPC que asocie con el dominio.

También puede crear una conexión de emparejamiento entre transit gateways de diferentes regiones de AWS. Esto le permite dirigir el tráfico entre las vinculaciones de las transit gateways a través de diferentes regiones.

Para obtener más información, consulte Transit gateways.

Acceso a un clúster de MemoryDB cuando este y la instancia de Amazon EC2 se encuentran en Amazon VPC diferentes en regiones distintas

Uso de la VPC de tránsito

Una alternativa a la utilización de la conexión de emparejamiento de VPC, otra estrategia común para conectar varias VPC y redes remotas dispersas geográficamente es crear una VPC de tránsito que sirva como un centro de tránsito de red global. Una VPC de tránsito simplifica la administración de la red y minimiza el número de conexiones necesarias para conectar varias VPC y redes remotas.

Este diseño puede ahorrar tiempo y esfuerzo, además de reducir los costos, ya que se implementa prácticamente sin los gastos tradicionales de establecer una presencia física en un hub de tránsito de coubicación o de implementar un equipo de red física.

Conexión entre VPC diferentes en distintas regiones

Una vez que la Amazon VPC de tránsito se encuentre establecida, se puede conectar una aplicación implementada en una VPC “radial” de una región a un clúster de MemoryDB de una VPC “radial” dentro de otra región.

Para obtener acceso a un clúster en una VPC diferente dentro de una región de AWS distinta

1. Implemente una solución de VPC de tránsito. Para obtener más información, consulte AWS Transit Gateway.
2. Actualice las tablas de enrutamiento de la VPC en la aplicación y las VPC para direccionar el tráfico a través de la VGW (gateway privada virtual) y el dispositivo de VPN. En caso de que se produzca el enrutamiento dinámico con el protocolo de gateway fronteriza (BGP), las rutas se pueden propagar automáticamente.
3. Modifique el grupo de seguridad de su clúster de MemoryDB para permitir la conexión de entrada del rango de IP de instancias de aplicación. Tenga en cuenta que no podrá remitirse al grupo de seguridad de servidor de la aplicación en este caso.

El acceso a un clúster entre regiones conllevará latencias de red y costos adicionales de transferencia de datos entre regiones.

Acceso a un clúster de MemoryDB desde una aplicación en ejecución en un centro de datos del cliente

Otra situación posible es una arquitectura híbrida en la que los clientes o las aplicaciones del centro de datos del cliente puedan necesitar obtener acceso a un clúster de MemoryDB en la VPC. Esta situación también se admite, siempre que haya conectividad entre la VPC del cliente y el centro de datos, ya sea a través de la VPN como de Direct Connect.

Temas

- Acceso a un clúster de MemoryDB desde una aplicación en ejecución en un centro de datos del cliente mediante conectividad de VPN
- Acceso a un clúster de MemoryDB desde una aplicación en ejecución en un centro de datos del cliente mediante Direct Connect

Acceso a un clúster de MemoryDB desde una aplicación en ejecución en un centro de datos del cliente mediante conectividad de VPN

Conexión a MemoryDB desde su centro de datos a través de una VPN

Para obtener acceso a un clúster en una VPC desde una aplicación local a través de una conexión de VPN

1. Para establecer la conectividad de VPN, agregue una gateway privada virtual de hardware a su VPC. Para obtener más información, consulte Adición de una gateway privada virtual de hardware a la VPC.
2. Actualiza la tabla de enrutamiento de VPC para la subred en la que se implementa su clúster de MemoryDB para permitir el tráfico desde el servidor de aplicaciones de sus instalaciones. En caso de que se produzca el enrutamiento dinámico con BGP, las rutas se pueden propagar automáticamente.
3. Modifique el grupo de seguridad de su clúster de MemoryDB para permitir la conexión de entrada desde los servidores de la aplicación en las instalaciones.

El acceso a un clúster a través de una conexión de VPN conllevará latencias de red y costos adicionales de transferencia de datos.

Acceso a un clúster de MemoryDB desde una aplicación en ejecución en un centro de datos del cliente mediante Direct Connect

Conexión a MemoryDB desde su centro de datos a través de Direct Connect

Para obtener acceso a un clúster de MemoryDB desde una aplicación en ejecución en su red mediante Direct Connect

1. Establezca la conectividad de Direct Connect. Para obtener más información, consulte Introducción a AWS Direct Connect.
2. Modifique el grupo de seguridad de su clúster de MemoryDB para permitir la conexión de entrada desde los servidores de la aplicación en las instalaciones.

El acceso a un clúster a través de una conexión de DX puede conllevar latencias de red y cargos adicionales por transferencia de datos.

Creación de una Virtual Private Cloud (VPC)

En este ejemplo, creará una nube privada virtual (VPC) basada en el servicio de Amazon VPC con una subred privada para cada zona de disponibilidad.

Creación de una VPC (consola)

Para crear un clúster de MemoryDB dentro de una Amazon Virtual Private Cloud

1. Inicie sesión en la consola de administración de AWS y abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.
2. En el panel de VPC, elija Create VPC (Crear VPC).
3. En Recursos para crear elija VPC y más.
4. En Number of Availability Zones (AZs) (Número de zonas de disponibilidad), seleccione el número de zonas de disponibilidad que quiere usar con las subredes.
5. En Number of public subnets (Número de subredes públicas), elija el número de subredes públicas que desea agregar a la VPC.
6. En Number of private subnets (Número de subredes privadas), elija el número de subredes públicas que desea agregar a la VPC.

   > **Tip**
   >
   > Anote los identificadores de las subredes e identifique cuál es pública y cuál es privada. Necesitará esta información más adelante al lanzar sus clústeres y agregar una instancia de Amazon EC2 a su Amazon VPC.

7. Cree un grupo de seguridad de Amazon VPC. Utilizará este grupo para su clúster y su instancia de Amazon EC2.

   a. En el panel de navegación izquierdo de la AWS Management Console, elija Grupos de seguridad.
   b. Elija Create Security Group.
   c. Introduzca un nombre y una descripción para el grupo de seguridad en los cuadros correspondientes. Para VPC, elija el identificador de su VPC.
   d. Una vez que la configuración sea la deseada, elija Yes, Create (Sí, crear).
8. Defina una regla de entrada de red para su grupo de seguridad. Esta regla permitirá conectarse a su instancia de Amazon EC2 mediante Secure Shell (SSH).

a. En el panel de navegación izquierdo, elija Security Groups.
b. Busque el grupo de seguridad en la lista y, a continuación, elíjalo.
c. En Security Group (Grupo de seguridad), elija la pestaña Inbound (Entrada). En el cuadro Create a new rule (Crear una nueva regla), elija SSH y, a continuación, elija Add Rule (Agregar regla).

   Establezca los siguientes valores para la regla de entrada nueva a fin de permitir el acceso HTTP:

   - Tipo: HTTP
   - Fuente: 0.0.0.0/0

d. Establezca los siguientes valores para la regla de entrada nueva a fin de permitir el acceso HTTP:

   - Tipo: HTTP
   - Fuente: 0.0.0.0/0

   Elija Apply Rule Changes (Aplicar cambios de regla).

Ahora se encuentra preparado para crear un grupo de subredes y crear un clúster en su VPC.

Subredes y grupos de subredes

Un grupo de subredes es una colección de subredes (que suelen ser privadas) que puede designar para los clústeres que se ejecutan en un entorno de Amazon Virtual Private Cloud (VPC).

Al crear un clúster en una Amazon VPC, pueden especificar un grupo de subredes o utilizar el grupo predeterminado que se proporciona. MemoryDB usa dicho grupo de subredes para elegir una subred y direcciones IP pertenecientes a dicha subred para asociarlas a sus nodos.

En esta sección se explica cómo crear y aprovechar las subredes y los grupos de subredes para administrar el acceso a los recursos de MemoryDB.

Para obtener más información sobre la utilización de grupos de subredes en entornos de Amazon VPC, consulte Paso 2: autorizar el acceso al clúster.

ID AZ de MemoryDB compatibles

Nombre de la región/ Región	ID AZ compatibles		
Región del este de EE. UU. (Ohio) `us-east-2`	`use2-az1, use2-az2, use2-az3`		
Región del este de EE. UU (N. Virginia) `us-east-1`	`use1-az2, use1-az4, use1-az6`		
Región del oeste de EE. UU (N. California) `us-west-1`	`usw1-az1, usw1-az2, usw1-az3`		
Región del oeste de EE. UU (Oregon) `us-west-2`	`usw2-az1, usw2-az2, usw2-az3`		
Canada (Central) Region	`cac1-az1, cac1-az2, cac1-az4`		

Nombre de la región/ Región	ID AZ compatibles		
ca-central-1			
Región Asia Pacífico (Hong Kong) ap-east-1	ape1-az1, ape1-az2, ape1-az3		
Región Asia-Pacífico (Mumbai) ap-south-1	aps1-az1, aps1-az2, aps1-az3		
Región Asia-Pacífico (Tokio) ap-northeast-1	apne1-az1, apne1-az2, apne1-az4		
Asia Pacific (Seoul) Region ap-northeast-2	apne2-az1, apne2-az2, apne2-az3		
Asia Pacific (Singapore) Region ap-southeast-1	apse1-az1, apse1-az2, apse1-az3		
Asia Pacific (Sydney) Region ap-southeast-2	apse2-az1, apse2-az2, apse2-az3		
Europe (Frankfurt) Region eu-central-1	euc1-az1, euc1-az2, euc1-az3		

Nombre de la región/ Región	ID AZ compatibles		
Europe (Ireland) Region `eu-west-1`	`euw1-az1, euw1-az2, euw1-az3`		
Europe (London) Region `eu-west-2`	`euw2-az1, euw2-az2, euw2-az3`		
Región Europa (Estocolmo) `eu-north-1`	`eun1-az1, eun1-az2, eun1-az3`		
South America (São Paulo) Region `sa-east-1`	sae1-az1, sae1-az2, sae1-az3		
Región China (Pekín) `cn-north-1`	cnn1-az1, cnn1-az2		
Región China (Ningxia) `cn-northwest-1`	cnw1-az1, cnw1-az2, cnw1-az3		

Temas

- Creación de un grupo de subredes
- Actualización de un grupo de subredes
- Visualización de los detalles de grupos de subredes
- Eliminación de un grupo de subredes

Creación de un grupo de subredes

Cuando cree un nuevo grupo de subredes, tenga en cuenta el número de direcciones IP disponibles. Si la subred tiene pocas direcciones IP libres, el número de nodos que podrá agregar al clúster será limitado. Para solucionar este problema, puede asignar una o varias subredes a un grupo de subredes para, de este modo, disponer de suficientes direcciones IP en la zona de disponibilidad de su clúster. Hecho esto, podrá agregar más nodos a su clúster.

En los siguientes procedimientos, se muestra cómo crear un grupo de subredes denominado `mysubnetgroup` (consola), la AWS CLI y la API de MemoryDB.

Creación de un grupo de subredes (consola)

En el siguiente procedimiento, se muestra cómo crear un grupo de subredes (consola).

Para crear un grupo de subredes (consola)

1. Inicie sesión en la consola de administración de AWS y abra la consola de MemoryDB en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación del lado izquierdo, elija Subnet Groups.
3. Elija Create Subnet Group (Crear grupo de subredes).
4. En la página Crear grupo de subredes, haga lo siguiente:

    a. En el cuadro Name (Nombre), escriba un nombre para el grupo de subredes.

        Las restricciones para la asignación de nombres de clúster son las siguientes:

        - Deben contener entre 1 y 40 caracteres alfanuméricos o guiones.
        - Deben comenzar por una letra.
        - No pueden contener dos guiones consecutivos.
        - No pueden terminar con un guion.

    b. En el cuadro Description(Descripción), escriba la descripción del grupo de subredes.

    c. En el cuadro de VPC ID (ID de la VPC), elija la Amazon VPC que creó anteriormente. Si no ha creado ninguna, pulse el botón Crear VPC y siga los pasos para crear una.

    d. En Subredes seleccionadas, elija la zona de disponibilidad y el ID de su subred privada y, a continuación, seleccione Elegir.

5. En Etiquetas, si lo desea, puede aplicar etiquetas para buscar y filtrar sus subredes o realizar un seguimiento de sus costos de AWS.

6. Cuando esté conforme con todos los ajustes, elija Crear.
7. En el mensaje de confirmación que aparece, elija Close (Cerrar).

El nuevo grupo de subredes aparecerá en la lista Grupos de subredes de la consola de MemoryDB. En la parte inferior de la ventana, podrá elegir el grupo de subredes para ver detalles tales como todas las subredes asociadas al grupo.

Creación de un grupo de subredes (CLI de AWS)

En el símbolo del sistema, utilice el comando `create-subnet-group` para crear un grupo de subredes.

Para Linux, macOS o Unix:

```
aws memorydb create-subnet-group \
    --subnet-group-name mysubnetgroup \
    --description "Testing" \
    --subnet-ids subnet-53df9c3a
```

Para Windows:

```
aws memorydb create-subnet-group ^
    --subnet-group-name mysubnetgroup ^
    --description "Testing" ^
    --subnet-ids subnet-53df9c3a
```

Este comando debería producir un resultado similar al siguiente:

```
    {
        "SubnetGroup": {
            "Subnets": [
                {
                    "Identifier": "subnet-53df9c3a",
                    "AvailabilityZone": {
                    "Name": "us-east-1a"
                    }
                }
            ],
            "VpcId": "vpc-3cfaef47",
            "Name": "mysubnetgroup",
```

```
            "ARN": "arn:aws:memorydb:us-east-1:012345678912:subnetgroup/
mysubnetgroup",
            "Description": "Testing"
        }
    }
```

Para obtener más información, consulte el tema de la AWS CLI create-subnet-group.

Creación de un grupo de subredes (API de MemoryDB)

Mediante la API de MemoryDB, realice una llamada a `CreateSubnetGroup` con los parámetros siguientes:

- `SubnetGroupName=`*`mysubnetgroup`*
- `Description=`*`Testing`*
- `SubnetIds.member.1=`*`subnet-53df9c3a`*

Actualización de un grupo de subredes

Puede actualizar la descripción de un grupo de subredes o modificar la lista de los ID de subred asociados al grupo de subredes. No puede eliminar un ID de subred desde un grupo de subredes si un clúster utiliza actualmente dicha subred.

Los procedimientos siguientes muestran cómo actualizar un grupo de subredes.

Actualización de grupos de subredes (consola)

Para actualizar un grupo de subredes

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación del lado izquierdo, elija Subnet Groups.
3. En la lista de grupos de subredes, elija el grupo que desea modificar.
4. Los campos Nombre, ID de VPC y Descripción no se pueden modificar.
5. En la sección Subredes seleccionadas, haga clic en Administrar para realizar cualquier cambio en las zonas de disponibilidad que necesite para las subredes. Para guardar los cambios, elija Save (Guardar).

Actualización de grupos de subredes (AWSCLI)

En el símbolo del sistema, utilice el comando `update-subnet-group` para actualizar un grupo de subredes.

Para Linux, macOS o Unix:

```
aws memorydb update-subnet-group \
    --subnet-group-name mysubnetgroup \
    --description "New description" \
    --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Para Windows:

```
aws memorydb update-subnet-group ^
    --subnet-group-name mysubnetgroup ^
    --description "New description" ^
    --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Este comando debería producir un resultado similar al siguiente:

```
{
    "SubnetGroup": {
        "VpcId": "vpc-73cd3c17",
        "Description": "New description",
        "Subnets": [
            {
                "Identifier": "subnet-42dcf93a",
                "AvailabilityZone": {
                    "Name": "us-east-1a"
                }
            },
            {
                "Identifier": "subnet-48fc12a9",
                "AvailabilityZone": {
                    "Name": "us-east-1a"
                }
            }
        ],
        "Name": "mysubnetgroup",
        "ARN": "arn:aws:memorydb:us-east-1:012345678912:subnetgroup/mysubnetgroup",
    }
}
```

Para obtener más información, consulte el tema de la AWS CLI update-subnet-group.

Actualización de grupos de subredes (API de MemoryDB)

Mediante la API de MemoryDB, realice una llamada a `UpdateSubnetGroup` con los parámetros siguientes:

- `SubnetGroupName=`*`mysubnetgroup`*
- Cualquier otro parámetro cuyos valores desea cambiar. Este ejemplo utiliza `Description=`*`New%20description`* para cambiar la descripción del grupo de subredes.

Example

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=UpdateSubnetGroup
    &Description=New%20description
```

```
    &SubnetGroupName=mysubnetgroup
    &SubnetIds.member.1=subnet-42df9c3a
    &SubnetIds.member.2=subnet-48fc21a9
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Timestamp=20141201T220302Z
    &Version=2014-12-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Credential=<credential>
    &X-Amz-Date=20141201T220302Z
    &X-Amz-Expires=20141201T220302Z
    &X-Amz-Signature=<signature>
    &X-Amz-SignedHeaders=Host
```

> **Note**
>
> Cuando cree un nuevo grupo de subredes, tenga en cuenta el número de direcciones IP disponibles. Si la subred tiene pocas direcciones IP libres, el número de nodos que podrá agregar al clúster será limitado. Para solucionar este problema, puede asignar una o varias subredes a un grupo de subredes para, de este modo, disponer de suficientes direcciones IP en la zona de disponibilidad de su clúster. Hecho esto, podrá agregar más nodos a su clúster.

Visualización de los detalles de grupos de subredes

Los procedimientos siguientes muestran cómo ver los detalles de un grupo de subredes.

Visualización de los detalles de los grupos de subredes (consola)

Para ver los detalles de un grupo de subredes (consola)

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación del lado izquierdo, elija Subnet Groups.
3. En la página Grupos de subredes, elija el grupo de subredes en Nombre o introduzca el nombre del grupo de subredes en la barra de búsqueda.
4. En la página Grupos de subredes, elija el grupo de subredes en Nombre o introduzca el nombre del grupo de subredes en la barra de búsqueda.

5. En Configuración del grupo de subredes, puede ver el nombre, la descripción, el ID de VPC y el nombre de recurso de Amazon (ARN) del grupo de subredes.
6. En Subredes, puede ver las zonas de disponibilidad, los ID de subred y los bloques CIDR del grupo de subredes
7. En Etiquetas, puede ver cualquier etiqueta asociada al grupo de subredes.

Visualización de los detalles de los grupos de subredes (AWSCLI)

En el símbolo del sistema, use el comando `describe-subnet-groups` para ver los detalles de un grupo de subredes específico.

Para Linux, macOS o Unix:

```
aws memorydb describe-subnet-groups \
    --subnet-group-name mysubnetgroup
```

Para Windows:

```
aws memorydb describe-subnet-groups ^
    --subnet-group-name mysubnetgroup
```

Este comando debería producir un resultado similar al siguiente:

```
{
  "subnetgroups": [
    {
      "Subnets": [
        {
          "Identifier": "subnet-060cae3464095de6e",
          "AvailabilityZone": {
            "Name": "us-east-1a"
          }
        },
        {
          "Identifier": "subnet-049d11d4aa78700c3",
          "AvailabilityZone": {
            "Name": "us-east-1c"
          }
        },
        {
```

```
          "Identifier": "subnet-0389d4c4157c1edb4",
          "AvailabilityZone": {
            "Name": "us-east-1d"
          }
        }
      ],
      "VpcId": "vpc-036a8150d4300bcf2",
      "Name": "mysubnetgroup",
      "ARN": "arn:aws:memorydb:us-east-1:53791xzzz7620:subnetgroup/mysubnetgroup",
      "Description": "test"
    }
  ]
}
```

Para ver los detalles de todos los grupos de subredes, utilice el mismo comando pero sin especificar un nombre de grupo de subredes.

```
aws memorydb describe-subnet-groups
```

Para obtener más información, consulte el tema de la AWS CLI describe-subnet-groups.

Visualización de grupos de subredes (API de MemoryDB)

Mediante la API de MemoryDB, realice una llamada a `DescribeSubnetGroups` con los parámetros siguientes:

`SubnetGroupName=mysubnetgroup`

Example

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=UpdateSubnetGroup
    &Description=New%20description
    &SubnetGroupName=mysubnetgroup
    &SubnetIds.member.1=subnet-42df9c3a
    &SubnetIds.member.2=subnet-48fc21a9
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Timestamp=20211801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
```

```
    &X-Amz-Credential=<credential>
    &X-Amz-Date=20210801T220302Z
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Signature=<signature>
    &X-Amz-SignedHeaders=Host
```

Eliminación de un grupo de subredes

Si decide que ya no necesita su grupo de subredes, puede eliminarlo. No puede eliminar un grupo de subredes si hay un clúster que lo utiliza actualmente. Tampoco se puede eliminar un grupo de subredes en un clúster con Multi-AZ habilitado si al hacerlo se deja ese clúster con menos de dos subredes. Primero debe desactivar Multi-AZ y, a continuación, eliminar la subred.

Los procedimientos que se describen a continuación muestran cómo eliminar un grupo de subredes.

Eliminación de un grupo de subredes (consola)

Para eliminar un grupo de subredes

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación del lado izquierdo, elija Subnet Groups.
3. En la lista de grupos de subredes, elija el grupo que desea eliminar, seleccione Acciones y, a continuación, Eliminar.

   > **Note**
   >
   > No puede eliminar un grupo de subredes predeterminado ni uno que esté asociado a ningún clúster.

4. Aparecerá la pantalla de confirmación Eliminar grupos de subredes confirmation screen will appear.
5. Para eliminar el grupo de subredes, introdúzcalo `delete` en el cuadro de texto de confirmación. Para mantener el grupo de subredes, seleccione Cancel (Cancelar).

Eliminación de un grupo de subredes (CLI de AWS)

Mediante la AWS CLI, llame al comando delete-subnet-group con el siguiente parámetro:

- `--subnet-group-name` *`mysubnetgroup`*

Para Linux, macOS o Unix:

```
aws memorydb delete-subnet-group \
```

```
    --subnet-group-name mysubnetgroup
```

Para Windows:

```
aws memorydb delete-subnet-group ^
    --subnet-group-name mysubnetgroup
```

Para obtener más información, consulte el tema de la AWS CLI delete-subnet-group.

Eliminación de un grupo de subredes (API de MemoryDB)

Mediante la API de MemoryDB, realice una llamada a `DeleteSubnetGroup` con el parámetro siguiente:

- `SubnetGroupName=mysubnetgroup`

Example

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DeleteSubnetGroup
    &SubnetGroupName=mysubnetgroup
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Timestamp=20210801T220302Z
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Credential=<credential>
    &X-Amz-Date=20210801T220302Z
    &X-Amz-Expires=20210801T220302Z
    &X-Amz-Signature=<signature>
    &X-Amz-SignedHeaders=Host
```

Este comando no genera ninguna salida.

Para obtener más información, consulte el tema de la API de MemoryDB DeleteSubnetGroup.

Puntos de conexión de la VPC de interfaz y API de MemoryDB para Redis (AWS PrivateLink)

Puede establecer una conexión privada entre los puntos de conexión de la VPC y la API de Amazon MemoryDB para Redis creando un punto de conexión de la VPC de tipo interfaz. Los puntos de

conexión de interfaz utilizan la tecnología de AWS PrivateLink. AWS PrivateLink le permite acceder de forma privada a las operaciones de la API de MemoryDB para Redis sin una puerta de enlace de Internet, un dispositivo NAT, una conexión VPN o una conexión AWS Direct Connect.

Las instancias de la VPC no necesitan direcciones IP públicas para comunicarse con los puntos de conexión de la API de MemoryDB para Redis. Las instancias tampoco necesitan direcciones IP públicas para utilizar ninguna de las operaciones de la API de MemoryDB disponibles. El tráfico entre la VPC y MemoryDB para Redis no sale de la red de Amazon. Cada punto de conexión de la interfaz está representado por una o más interfaces de red elásticas en las subredes. Para obtener más información sobre las interfaces de red elásticas, consulte Interfaces de red elásticas en la Guía del usuario de Amazon EC2.

- Para obtener más información sobre puntos de conexión de la VPC, consulte Puntos de conexión de la VPC de tipo interfaz (AWS PrivateLink) en la Guía del usuario de Amazon VPC.
- Para obtener más información sobre las operaciones de la API de MemoryDB, consulte Operaciones de la API de MemoryDB.

Después de crear un punto de conexión de la VPC de tipo interfaz, si habilita nombres de host de DNS privados para el punto de conexión, el punto de conexión predeterminado de MemoryDB (https://memorydb.*Region*.amazonaws.com) se resuelve en el punto de conexión de la VPC. Si no habilita nombres de host de DNS privados, Amazon VPC proporciona un nombre de punto de conexión de DNS que puede utilizar en el siguiente formato:

```
VPC_Endpoint_ID.memorydb.Region.vpce.amazonaws.com
```

Para obtener más información, consulte Puntos de conexión de la VPC de tipo interfaz (AWS PrivateLink) en la Guía del usuario de Amazon VPC. MemoryDB permite realizar llamadas a todas sus acciones de API dentro de su VPC.

> **Note**
>
> Los nombres de host DNS privados solo se pueden habilitar para un punto de conexión de VPC en la VPC. Si quiere crear un punto de conexión de VPC adicional, el nombre de host DNS privado debe estar deshabilitado para ello.

Consideraciones para los puntos de conexión de VPC de

Antes de configurar un punto de conexión de la VPC de tipo interfaz para los puntos de conexión de la API de MemoryDB para Redis, asegúrese de revisar Propiedades y limitaciones de puntos de conexión de interfaz en la Guía del usuario de Amazon VPC. Todas las operaciones de la API de MemoryDB relevantes para la administración de MemoryDB para Redis están disponibles desde la VPC mediante el uso de AWS PrivateLink. Las políticas de puntos de conexión de VPC son compatibles con los puntos de conexión de la API de MemoryDB. De forma predeterminada, se permite el acceso completo a las operaciones de la API de MemoryDB a través del punto de conexión. Para obtener más información, consulte Control del acceso a los servicios con puntos de conexión de la VPC en la Guía del usuario de Amazon VPC.

Creación de un punto de conexión de la VPC de interfaz para la API de MemoryDB

Puede crear un punto de conexión de VPC para la API de MemoryDB para Redis mediante la consola de Amazon VPC o la AWS CLI. Para obtener más información, consulte Creación de un punto de conexión de interfaz en la Guía del usuario de Amazon VPC.

Después de crear un punto de enlace de la VPC de interfaz, puede habilitar nombres de host de DNS privados para el punto de conexión. Cuando lo haga, el punto de conexión predeterminado de MemoryDB para Redis (https://memorydb.*Región*.amazonaws.com) se resuelve en el punto de conexión de VPC. Para obtener más información, consulte Acceso a un servicio a través de un punto de conexión de interfaz en la Guía del usuario de Amazon VPC.

Creación de una política de punto de conexión de VPC para la API de Amazon MemoryDB

Puede asociar una política de punto de conexión al punto de conexión de la VPC que controla el acceso a la API de MemoryDB. La política especifica lo siguiente:

- La entidad de seguridad que puede realizar acciones.
- Las acciones que se pueden realizar.
- Los recursos en los que se pueden llevar a cabo las acciones.

Para obtener más información, consulte Control del acceso a los servicios con puntos de conexión de la VPC en la Guía del usuario de Amazon VPC.

Example Política de punto de conexión de la VPC para acciones de la API de MemoryDB

A continuación, se muestra un ejemplo de una política de punto de conexión para la API de MemoryDB. Cuando se asocia a un punto de conexión, esta política concede acceso a las acciones de la API de MemoryDB enumeradas para todos las entidades principales de todos los recursos.

```
{
 "Statement": [{
  "Principal": "*",
  "Effect": "Allow",
  "Action": [
   "memorydb:CreateCluster",
   "memorydb:UpdateCluster",
   "memorydb:CreateSnapshot"
  ],
  "Resource": "*"
 }]
}
```

Example Política de punto de conexión de VPC que deniega todo el acceso desde una cuenta de AWS especificada

La siguiente política de punto de conexión de VPC deniega a la cuenta de AWS *123456789012* todo el acceso a los recursos mediante el punto de conexión. La política permite todas las acciones de otras cuentas.

```
{
 "Statement": [{
   "Action": "*",
   "Effect": "Allow",
   "Resource": "*",
   "Principal": "*"
  },
  {
   "Action": "*",
   "Effect": "Deny",
   "Resource": "*",
   "Principal": {
    "AWS": [
     "123456789012"
    ]
   }
```

```
  }
 ]
}
```

Actualizaciones de los servicios de MemoryDB para Redis

MemoryDB para Redis monitorea automáticamente la flota de clústeres y nodos de Redis para aplicar actualizaciones de servicio cuando se encuentran disponibles. Normalmente, se configura un periodo de mantenimiento predefinido para que MemoryDB pueda aplicar estas actualizaciones. Sin embargo, en algunos casos es posible que este enfoque resulte demasiado rígido y que probablemente restrinja los flujos de negocio.

Con las actualizaciones de servicio, usted controla cuándo y qué actualizaciones se aplican. También puede monitorear el progreso de estas actualizaciones conforme se aplican a los clústeres de MemoryDB seleccionados en tiempo real.

Administración de las actualizaciones de servicio

Las actualizaciones del servicio MemoryDB se publican periódicamente. Si tiene uno o varios clústeres que reúnen los requisitos de estas actualizaciones, recibirá notificaciones por correo electrónico, SNS, Personal Health Dashboard (PHD) y eventos de Amazon CloudWatch cuando se seleccionan las actualizaciones. Las actualizaciones se muestran también en la página Actualizaciones de servicio de la consola de MemoryDB. Mediante este panel, puede ver todas las actualizaciones de servicio y su estado para su flota MemoryDB.

Puede controlar cuándo se debe aplicar una actualización antes de que se inicie una actualización automática. Es absolutamente recomendable aplicar las actualizaciones del tipo security-update lo antes posible para asegurarse de que su MemoryDB esté siempre actualizado con los parches de seguridad actuales.

En las siguientes secciones se describen detalladamente las opciones.

Temas

Aplicación de las actualizaciones de servicio

Puede comenzar a aplicar las actualizaciones del servicio a la flota Redis desde el momento en que las actualizaciones tengan el estado available (disponible). Las actualizaciones del servicio son

acumulativas. Es decir, todas las actualizaciones que no se hayan aplicado se incluirán con la última actualización.

Si una actualización de servicio tiene habilitada la actualización automática, puede optar por no realizar ninguna acción cuando esté disponible. MemoryDB programará la aplicación de la actualización durante el periodo de mantenimiento de los clústeres después de la fecha de inicio de la actualización automática. Recibirá notificaciones relacionadas con cada etapa de la actualización.

> **Note**
>
> Solo puede aplicar las actualizaciones de servicio que tengan un estado disponible o programado.

Para obtener más información sobre cómo revisar y aplicar actualizaciones específicas del servicio a los clústeres de MemoryDB correspondientes, consulte Aplicación de las actualizaciones de servicio con la consola.

Cuando una actualización nueva del servicio está disponible para uno o varios clústeres de MemoryDB, puede utilizar la consola de MemoryDB, la API o la AWS CLI para aplicar la actualización. En las siguientes secciones se explican las opciones que puede utilizar para aplicar las actualizaciones.

Aplicación de las actualizaciones de servicio con la consola

Para consultar la lista de las distintas actualizaciones de servicio disponibles, junto con otra información, vaya a Service Updates (Actualizaciones de servicio) en la consola.

1. Inicie sesión en la AWS Management Console y abra la consola de MemoryDB para Redis en https://console.aws.amazon.com/memorydb/.
2. En el panel de navegación, seleccione Service Updates (Actualizaciones de servicio).

En Detalles de la actualización del servicio puede ver lo siguiente:

- Service update name (Nombre de actualización de servicio): el nombre único de la actualización de servicio
- Descripción de la actualización: proporciona información detallada sobre la actualización del servicio

- Fecha de inicio de la actualización automática: si se establece este atributo, MemoryDB empezará a programar sus clústeres para que se actualicen automáticamente en los periodos de mantenimiento correspondientes después de esta fecha. Recibirá notificaciones por adelantado en el periodo exacto de mantenimiento programado, que puede no ser el inmediatamente posterior a la fecha de inicio de la actualización automática. Puede seguir aplicando la actualización a sus clústeres en cualquier momento que desee. Si el atributo no está establecido, la actualización del servicio no está habilitada para la actualización automática y MemoryDB no actualizará los clústeres automáticamente.

En la sección Cluster update status (Estado de actualización del clúster), puede ver una lista de clústeres en los que la actualización del servicio no se ha aplicado o se ha aplicado recientemente. Para cada clúster, puede ver lo siguiente:

- Cluster name (Nombre del clúster): el nombre del clúster
- Nodes Updated (Nodos actualizados): la proporción de nodos en un clúster específico que se actualizaron o que permanecen disponibles para la actualización del servicio específica.
- Update Type (Tipo de actualización): el tipo de actualización de servicio, que es security-update o engine-update
- Status (Estado): el estado de la actualización de servicio en el clúster, que es uno de los siguientes:
  - available (disponible): la aplicación está lista para los clústeres Redis correspondientes.
  - in-progres (en progreso): la actualización se está aplicación a este clúster.
  - scheduled (programado): se ha programado la fecha de actualización.
  - complete (completa): la actualización se ha aplicado correctamente. El clúster con el estado completo se mostrará durante 7 días después de su finalización.

  Si ha elegido alguno o todos los clústeres con estado available (disponible) o scheduled (programado) y, luego, eligió Apply now (Postúlese ahora), la actualización empezará a aplicarse en esos clústeres.

Aplicación de las actualizaciones de servicio con la AWS CLI

Tras recibir una notificación de que hay actualizaciones del servicio disponibles, puede inspeccionarlas y aplicarlas con AWS CLI:

- Para recuperar una descripción de las actualizaciones de servicio disponibles, ejecute el siguiente comando:

  ```
  aws memorydb describe-service-updates --status available
  ```

  Para obtener más información, consulte describe-service-updates.
- Para aplicar una actualización de servicio en una lista de clústeres, ejecute el siguiente comando:

  ```
  aws memorydb batch-update-cluster --service-update
  ServiceUpdateNameToApply=sample-service-update --cluster-names cluster-1
  cluster2
  ```

  Para obtener más información, consulte batch-update-cluster.

Referencia

En los temas de esta sección, se explica cómo se trabaja con la API de MemoryDB y la sección sobre MemoryDB de la AWS CLI. También se describen mensajes de error y notificaciones de servicio comunes.

- Uso de la API de MemoryDB
- Referencia de la API de MemoryDB
- Sección MemoryDB de la Referencia AWS CLI

Uso de la API de MemoryDB

Esta sección proporciona descripciones orientadas a tareas acerca del uso y la implementación de operaciones de MemoryDB. Para obtener una descripción completa de dichas operaciones, consulte la Referencia de la API de MemoryDB.

Temas

Uso de la API de consultas

Parámetros de consulta

Las solicitudes basadas en consultas HTTP son solicitudes HTTP que utilizan el verbo HTTP GET o POST y un parámetro de consulta denominado `Action`.

Cada solicitud de consulta debe incluir algunos parámetros comunes para realizar la autenticación y la selección de una acción.

Algunas operaciones toman listas de parámetros. Estas listas se especifican utilizando la notación `param.`*n*. Los valores de *n* son números enteros a partir de 1.

Autenticación de solicitudes de consulta

Solo se pueden enviar solicitudes de consulta a través de HTTPS y cada una de ellas debe incluir una firma. En esta sección se describe cómo crear la firma. El método que se describe en el procedimiento siguiente se conoce como firma versión 4.

A continuación se indican los pasos básicos que se utilizan para autenticar las solicitudes en AWS. En este proceso se presupone que se ha registrado en AWS y que dispone de un ID de clave de acceso y una clave de acceso secreta.

Proceso de autenticación de consulta

1. El remitente crea una solicitud para AWS.

2. El remitente calcula la firma de la solicitud, una operación hash para código de autenticación de mensajes (HMAC) basado en hash mediante una función hash SHA-1, tal y como se define en la siguiente sección de este tema.
3. El remitente de la solicitud envía a AWS los datos de la misma, la firma y el ID de clave de acceso (el identificador de clave de la clave de acceso secreta utilizada).
4. AWS utiliza el ID de clave de acceso para buscar la clave de acceso secreta.
5. AWS genera una firma a partir de los datos de la solicitud y la clave de acceso secreta con el mismo algoritmo que se utilizó para calcular la firma de la solicitud.
6. Si las firmas coinciden, se considera que la solicitud es auténtica. Si la comparación falla, se descarta la solicitud y AWS devuelve una respuesta de error.

> **Note**
>
> Si una solicitud contiene un parámetro `Timestamp`, la firma calculada para la solicitud caduca 15 minutos después de su valor.
> Si una solicitud contiene un parámetro `Expires`, la firma caduca en el momento especificado por el parámetro `Expires`.

Para calcular la firma de la solicitud

1. Cree la cadena de consulta canónica que necesitará más adelante en este procedimiento:
    a. Ordene los componentes UTF-8 de la cadena de consulta por nombre de parámetro con el orden de bytes natural. Los parámetros pueden proceder de GET URI o del cuerpo de la solicitud POST (cuando Content-Type es application/x-www-form-urlencoded).
    b. Codifique como dirección URL el nombre y los valores del parámetro, aplicando las reglas siguientes:
        i. No incluya en la codificación de la dirección URL ninguno de los caracteres no reservados definidos en la norma RFC 3986. Estos caracteres no reservados son A–Z, a–z, 0–9, guion (-), carácter de subrayado (_), punto (.) y tilde (~).
        ii. Codifique con signos de porcentaje el resto de los caracteres con %XY, donde X e Y son caracteres hexadecimales (0-9 y A-F mayúsculas).
        iii. Codifique con signos de porcentaje los caracteres extendidos UTF-8 con el formato %XY%ZA...

iv. Codifique con el signo de porcentaje el carácter de espacio como %20 y no como + (lo que se hace en las codificaciones comunes).

c. Separe los nombres de los parámetros codificados de sus valores codificados con el signo igual (=) (carácter ASCII 61), aunque el valor del parámetro esté vacío.

d. Separe los pares de nombre-valor con el carácter ampersand (&) (código ASCII 38).

2. Cree la cadena para firmar según la siguiente pseudogramática ("\n" representa un carácter de nueva línea ASCII).

```
StringToSign = HTTPVerb + "\n" +
ValueOfHostHeaderInLowercase + "\n" +
HTTPRequestURI + "\n" +
CanonicalizedQueryString <from the preceding step>
```

El componente HTTPRequestURI es el componente de la ruta absoluta HTTP del URI hasta la cadena de consulta, pero sin incluirla. Si HTTPRequestURI está vacío, utilice una barra diagonal (/).

3. Calcule una HMAC conforme con RFC 2104 con la cadena que acaba de crear, su clave de acceso secreta como la clave y SHA256 o SHA1 como algoritmo de hash.

Para obtener más información, consulte https://www.ietf.org/rfc/rfc2104.txt.

4. Convierta el valor resultante en base 64.

5. Incluya el valor como valor del parámetro `Signature` de la solicitud.

A continuación se muestra una solicitud de muestra (se han agregado saltos de línea para facilitar la lectura).

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DescribeClusters
    &ClusterName=myCluster
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Version=2021-01-01
```

Para la cadena de consulta anterior, debería calcular la firma HMAC de la siguiente cadena.

```
GET\n
    memory-db.amazonaws.com\n
    Action=DescribeClusters
    &ClusterName=myCluster
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE%2F20140523%2Fus-east-1%2Fmemorydb%2Faws4_request
    &X-Amz-Date=20210801T223649Z
    &X-Amz-SignedHeaders=content-type%3Bhost%3Buser-agent%3Bx-amz-content-sha256%3Bx-
amz-date
        content-type:
        host:memory-db.us-east-1.amazonaws.com
        user-agent:ServicesAPICommand_Client
    x-amz-content-sha256:
    x-amz-date:
```

El resultado es la siguiente solicitud firmada.

```
https://memory-db.us-east-1.amazonaws.com/
    ?Action=DescribeClusters
    &ClusterName=myCluster
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &Version=2021-01-01
    &X-Amz-Algorithm=Amazon4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE/20141201/us-east-1/memorydb/aws4_request
    &X-Amz-Date=20210801T223649Z
    &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
    &X-Amz-Signature=2877960fced9040b41b4feaca835fd5cfeb9264f768e6a0236c9143f915ffa56
```

Para obtener información detallada acerca del proceso de firma y el cálculo de la firma de la solicitud, consulte el tema Proceso de firma Signature Version 4 y sus subtemas.

Bibliotecas disponibles

AWS ofrece kits de desarrollo de software (SDK) para los desarrolladores de software que prefieran crear aplicaciones mediante las API de lenguaje específico en lugar de la API de consulta. Estos SDK proporcionan funciones básicas (que no se incluyen en las API), como la autenticación de

solicitudes, los reintentos de solicitudes y la gestión de errores para que se pueda comenzar más fácilmente. Encontrará SDK y recursos adicionales para los siguientes lenguajes de programación:

- Java
- Windows y .NET
- PHP
- Python
- Ruby

Para obtener información acerca de otros lenguajes, consulte Código de muestra y bibliotecas.

Solución de problemas de aplicaciones

MemoryDB proporciona errores específicos y descriptivos para ayudarle a solucionar problemas durante la interacción con la API de MemoryDB.

Recuperación de errores

Normalmente, conviene que una aplicación compruebe si una solicitud generó un error antes de emplear tiempo en procesar los resultados. La forma más fácil de averiguar si se ha producido un error, consiste en buscar un nodo `Error` en la respuesta de la API de MemoryDB.

La sintaxis XPath permite comprobar fácilmente si hay un nodo `Error`, y ofrece un método sencillo de recuperar el mensaje de error y su código. La partición de código siguiente utiliza Perl y el módulo XML::XPath para determinar si se ha producido un error durante una solicitud. Si es así, el código imprime el primer mensaje de error y su código en la respuesta.

```
use XML::XPath;
my $xp = XML::XPath->new(xml =>$response);
if ( $xp->find("//Error") )
{print "There was an error processing your request:\n", " Error code: ",
$xp->findvalue("//Error[1]/Code"), "\n", " ",
$xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

Consejos para la solución de problemas

Recomendamos los siguientes procesos para diagnosticar y solucionar problemas con la API de MemoryDB.

- Compruebe que MemoryDB se está ejecutando correctamente.

  Para ello, solo tiene que abrir una ventana del navegador y enviar una solicitud de consulta al servicio de MemoryDB (como https://memory-db.us-east-1.amazonaws.com). Los mensajes MissingAuthenticationTokenException o UnknownOperationException confirman que el servicio está disponible y que responde a las solicitudes.
- Comprobar la estructura de la solicitud.

  Cada operación de MemoryDB tiene una página de referencia en la Referencia de la API de MemoryDB. Compruebe que utiliza los parámetros correctamente. Para obtener ideas sobre lo que podría estar mal, examine las solicitudes de muestra o los escenarios de usuario para ver si esos ejemplos realizan operaciones similares.
- Visite el foro.

  Existe un foro de debate de MemoryDB donde puede buscar soluciones a los problemas que otras personas han experimentado al usar este servicio. Para ver el foro, consulte

  https://forums.aws.amazon.com/

Cuotas para Amazon MemoryDB para Redis

Tu AWS cuenta tiene cuotas predeterminadas, antes denominadas límites, para cada AWS servicio. A menos que se indique lo contrario, cada cuota es específica de la región de . Puede solicitar el aumento de algunas cuotas, pero otras no se pueden aumentar.

Para solicitar un aumento de cuota, consulte Solicitud de aumento de cuota en la Guía del usuario de Service Quotas. Si la cuota aún no se encuentra disponible en Service Quotas, utilice el formulario de aumento del límite.

Su AWS cuenta tiene las siguientes cuotas relacionadas con MemoryDB.

Recurso	Predeterminado
Nodos por región	300
Nodos por clúster por tipo de instancias	90
Nodos por partición	6
Grupos de parámetros por región	150
Grupos de subredes por región	150
Subredes por grupo de subredes	20
Usuarios por grupo de usuarios	100
Número total de usuarios	1 000
Número de grupos de usuarios	100

Historial de documentos de la Guía del usuario de MemoryDB

En la siguiente tabla se describen las versiones de la documentación de MemoryDB.

Cambio	Descripción	Fecha
MemoryDB ahora es compatible con la autenticación de usuarios mediante IAM	La autenticación de IAM le permite autenticar una conexión a MemoryDB para Redis mediante identidades de AWS Identity and Access Management. Esto le permite reforzar el modelo de seguridad y simplificar muchas tareas de seguridad administrativa. Para obtener más información, consulte Authenticating with IAM (Autenticación con IAM).	10 de mayo de 2023
MemoryDB ahora admite Redis 7	Esta versión incorpora varias funciones nuevas a MemoryDB para Redis: funciones de Redis, mejoras de ACL y publicación/envío fragmentado y multiplexación de E/S mejorada. Para obtener más información, consulte Versiones del motor de Redis.	9 de mayo de 2023
MemoryDB ahora ofrece nodos reservados	Los nodos reservados ofrecen un descuento importante en comparación con los precios de los nodos bajo demanda.	27 de diciembre de 2022

	Los nodos reservados no son nodos físicos, sino más bien un descuento de facturación que se aplica al uso de nodos bajo demanda en su cuenta. Para obtener más información, consulte Nodos reservados de MemoryDB.	
MemoryDB ahora admite la organización de datos en niveles	Organización de datos en niveles de MemoryDB para Redis. Puede utilizar la organización de datos en niveles como una forma más económica de escalar los clústeres hasta cientos de terabytes de capacidad. Para obtener más información, consulte Organización de datos en niveles.	3 de noviembre de 2022

MemoryDB ahora admite el formato JavaScript Object Notation (JSON) nativo	El formato nativo de JavaScript Object Notation (JSON, Notación de objetos de JavaScript) es una forma sencilla y sin esquemas de codificar conjuntos de datos complejos dentro de clústeres de Redis. Puede almacenar datos de forma nativa y acceder a ellos utilizando el formato JSON dentro de clústeres de Redis, así como actualizar los datos JSON almacenados en esos clústeres, sin necesidad de administrar un código personalizado para serializarlo y deserializarlo. Para obtener más información, consulte Introducción a JSON.	25 de mayo de 2022
MemoryDB ahora es compatible con PrivateLink AWS	AWS PrivateLink permite obtener acceso de forma privada a las operaciones de la API de MemoryDB sin una puerta de enlace de Internet, un dispositivo NAT, una conexión VPN o una conexión AWS Direct Connect. Para obtener más información, consulte Puntos de conexión de la VPC de tipo interfaz y la API de MemoryDB (AWS PrivateLink).	24 de enero de 2022

Versión inicial	Versión inicial de la Guía del usuario de MemoryDB. Para obtener más información, consulte ¿Qué es MemoryDB? .	19 de agosto de 2021

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.