



Guida per gli sviluppatori

Amazon MemoryDB per Redis



Amazon MemoryDB per Redis: Guida per gli sviluppatori

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Cos'è MemoryDB per Redis?	1
Caratteristiche di MemoryDB	1
Componenti core MemoryDB	2
Cluster	3
Nodi	4
Shard	4
Gruppi di parametri	5
Gruppi di sottoreti	5
Liste di controllo accessi	5
Utenti	6
Servizi correlati	6
Scelta di regioni e zone di disponibilità	6
Ubicazione dei nodi	8
Regioni ed endpoint supportati	9
Accesso a MemoryDB	12
Sicurezza di MemoryDB	13
Guida introduttiva a MemoryDB	14
Configurazione	14
Crea il tuo account AWS	15
Concessione dell'accesso programmatico	16
Configura le tue autorizzazioni (solo nuovi utenti di MemoryDB)	18
Scaricamento e configurazione della CLI AWS	19
Fase 1: creazione di un cluster	20
Creazione di un cluster MemoryDB	20
Configurazione dell'autenticazione	31
Passaggio 2: autorizzare l'accesso al cluster	32
Fase 3: Connect al cluster	34
Trova il tuo endpoint del cluster	34
Connect a un cluster MemoryDB (Linux)	34
Fase 4: Eliminazione di un cluster	36
A questo punto come si può procedere?	38
Gestione di nodi	39
Nodi e shard di MemoryDB	39
Tipi di nodi supportati	41

Nodi riservati	43
Panoramica dei nodi riservati	43
Sostituzione dei nodi	54
Gestione dei cluster	57
Tiering di dati	58
Best practice	59
Restrizioni	59
Prezzi del tiering di dati	60
Monitoraggio	60
Utilizzo del tiering di dati	60
Ripristino di dati da da da da un'istantanea su cluster con tiering di dati abil(tiering di dati abil(tiering di dati abil	62
Preparazione di un cluster	63
Determina i tuoi requisiti	64
Creazione di un cluster	67
Visualizzazione dei dettagli di un cluster	68
Modifica di un cluster	73
Aggiunta/rimozione di nodi da un cluster	76
Accesso al cluster	78
Concedere l'accesso al cluster	78
Accesso a MemoryDB dall'esternoAWS	80
Individuazione degli endpoint di connessione	86
Partizioni	89
Ricerca	90
Gestione dell'implementazione di MemoryDB	94
Versioni del motore	94
Redis 7.0 (avanzata)	94
Redis 7.0 (avanzata)	95
Redis 6.2 (avanzato)	96
Aggiornamento delle versioni del motore	97
Nozioni di base di JSON	99
Panoramica del tipo di dati Redis JSON	100
Comandi supportati	112
Tagging delle risorse MemoryDB	154
Monitoraggio dei costi con i tag	159
Gestione dei tag con la AWS CLI	161

Gestione dei tag tramite l'API MemoryDB	164
Gestione della manutenzione	166
Best practice	168
Comandi Redis limitati	169
Resilienza	170
Best practice: Pub/Sub e multiplexdi I/O avanzato	172
Best practice. Dimensionamento di cluster online	172
Informazioni sulla replica MemoryDB	173
Consistenza	174
Replica in un cluster	174
Ridurre al minimo i tempi di inattività con la funzione Multi-AZ	175
Modifica del numero di repliche	183
Snapshot e ripristino	193
Vincoli	194
Costi	194
Pianificazione di istantanee automatiche	195
Creazione di istantanee manuali	196
Creazione di un'istananea finale	199
Descrizione delle istantanee	201
Copia di uno snapshot	204
Esportazione di un'istananea	207
Ripristino da uno snapshot	217
Seminare un cluster con un'istananea	223
Taggare le istantanee	229
Eliminazione di uno snapshot	230
Dimensionamento	231
Dimensionamento di cluster di MemoryDB	233
Configurazione dei parametri di motore con i gruppi di parametri	255
Gestione dei parametri	257
Livelli dei gruppi di parametri	258
Creazione di un gruppo di parametri	259
Elenco di gruppi di parametri per nome	264
Generazione di un elenco di valori di un gruppo di parametri	269
Modifica di un gruppo di parametri	270
Eliminazione di un gruppo di parametri	273
Parametri specifici di Redis	275

Tutorial: Configurazione di una funzione Lambda per accedere a MemoryDB in un Amazon VPC	292
Fase 1: creazione di un cluster	292
Passaggio 2: creazione di una funzione Lambda	295
Fase 3: esecuzione del test della funzione Lambda	299
Fase 4: Pulizia (opzionale)	300
Ricerca vettoriale	302
Panoramica della ricerca vettoriale	302
Indici e spazi chiave	303
Tipi di campi indice	304
Algoritmi per indici vettoriali	305
Espressione di interrogazione di ricerca vettoriale	306
Comando INFO	308
Sicurezza della ricerca vettoriale	311
Caratteristiche e limiti della ricerca vettoriale	311
Disponibilità della ricerca vettoriale	312
Restrizioni parametriche	312
Limiti di scalabilità	313
Restrizioni operative	313
Importazione/esportazione di istantanee e Live Migration	313
Consumo di memoria	314
Memoria insufficiente durante il riempimento	314
Transazioni	314
Casi d'uso	314
Retrieval Augmented Generation (RAG)	314
Memoria buffer Foundation Model (FM)	315
Rilevamento di attività fraudolente	316
Altri casi d'uso	316
Utilizzando il AWS Management Console	317
Utilizzando il AWS Command Line Interface	317
Comandi di ricerca vettoriale	318
FT.CREATE	318
FT.SEARCH	323
FT.AGGREGATE	325
FT.DROPINDEX	327
FT.INFO	327

FT. _LISTA	330
FT.ALIASADD	330
FT.ALIASDEL	330
FT.ALIASUPDATE	330
FT. _LISTA DI ALIAS	331
FT.CONFIG OTTIENI	331
AIUTO FT.CONFIG	331
SET FT.CONFIG	332
FT.PROFILE	332
FT.EXPLAIN	333
FT.EXPLAINCLI	333
Sicurezza	334
Protezione dei dati	335
Sicurezza dei dati in MemoryDB per Redis	336
Crittografia dei dati inattivi	337
Crittografia dei dati in transito (TLS)	339
Autenticazione degli utenti con ACL	340
Autenticazione con IAM	355
Gestione dell'identità e degli accessi	363
Destinatari	363
Autenticazione con identità	364
Gestione dell'accesso con policy	368
Come funziona MemoryDB for Redis con IAM	370
Esempi di policy basate su identità	381
Risoluzione dei problemi	384
Controllo accessi	386
Panoramica sulla gestione degli accessi	387
Registrazione di log e monitoraggio	416
Monitoraggio con CloudWatch	417
Monitoraggio degli eventi	437
Registrazione delle chiamate API MemoryDB per Redis conAWS CloudTrail	451
Convalida della conformità	458
Sicurezza dell'infrastruttura	459
Riservatezza del traffico Internet	459
MemoryDB e Amazon VPC	460
Sottoreti e gruppi di sottoreti	472

API MemoryDB per l'API Redis ed endpoint VPC di interfaccia (AWS PrivateLink)	486
Aggiornamenti del servizio	489
Gestire gli aggiornamenti del servizio	490
Riferimento	493
Utilizzo dell'API MemoryDB	494
Uso dell'API query	494
Librerie disponibili	497
Risoluzione dei problemi delle applicazioni	498
Quote	500
Cronologia dei documenti	501
.....	div

Cos'è MemoryDB per Redis?

MemoryDB for Redis è un servizio di database in memoria durevole che offre prestazioni ultraveloci. È progettato appositamente per applicazioni moderne con architetture di microservizi.

MemoryDB è compatibile con Redis, un popolare data store open source, che consente di creare rapidamente applicazioni utilizzando le stesse strutture di dati, API e comandi Redis flessibili e intuitivi che già utilizzano oggi. Con MemoryDB, tutti i dati vengono archiviati in memoria, il che consente di ottenere una latenza di lettura di microsecondi e una latenza di scrittura di millisecondi a una cifra e un throughput elevato. MemoryDB archivia inoltre i dati in modo duraturo su più zone di disponibilità (AZ) utilizzando un log transazionale Multi-AZ per consentire il failover rapido, il ripristino del database e il riavvio dei nodi.

Offrendo prestazioni in memoria e durabilità Multi-AZ, MemoryDB può essere utilizzato come database primario ad alte prestazioni per le applicazioni di microservizi, eliminando la necessità di gestire separatamente sia una cache che un database durevole.

Argomenti

- [Caratteristiche di MemoryDB](#)
- [Componenti core MemoryDB](#)
- [Servizi correlati](#)
- [Scelta di regioni e zone di disponibilità](#)
- [Accesso a MemoryDB](#)
- [Sicurezza di MemoryDB](#)

Caratteristiche di MemoryDB

MemoryDB for Redis è un servizio di database in memoria durevole che offre prestazioni ultraveloci. Le caratteristiche di MemoryDB includono:

- Forte coerenza per i nodi primari e coerenza finale garantita per i nodi di replica. Per ulteriori informazioni, consulta [Consistenza](#).
- Latenze di lettura in microsecondi e di scrittura a una cifra di millisecondi con un massimo di 160 milioni di TPS per cluster.

- Strutture dati e API Redis flessibili e intuitive. Crea facilmente nuove applicazioni o migra le applicazioni Redis esistenti con quasi nessuna modifica.
- Durabilità dei dati tramite un registro transazionale Multi-AZ che consente il ripristino e il riavvio rapidi del database.
- Disponibilità Multi-AZ con failover automatico, rilevamento e ripristino dei guasti dei nodi.
- Scala facilmente orizzontalmente aggiungendo e rimuovendo nodi o verticalmente passando a tipi di nodi più grandi o più piccoli. È possibile scalare il throughput di scrittura aggiungendo shard e scalare il throughput di lettura aggiungendo repliche.
- Read-after-write Consistenza R per i nodi primari e coerenza finale garantita per i nodi di replica.
- MemoryDB supporta la crittografia in transito, la crittografia a riposo e l'autenticazione degli utenti tramite. [Autenticazione degli utenti con elenchi di controllo degli accessi \(ACL\)](#)
- Istantanee automatiche in Amazon S3 con conservazione fino a 35 giorni.
- Support per un massimo di 500 nodi e più di 100 TB di storage per cluster (con 1 replica per shard).
- Crittografia in transito con TLS e crittografia inattiva con chiavi. AWS KMS
- Autenticazione e autorizzazione degli utenti con Redis. [Autenticazione degli utenti con elenchi di controllo degli accessi \(ACL\)](#)
- Support per i tipi di AWS istanze Graviton2.
- Integrazione con altri AWS servizi come CloudWatch Amazon VPC e Amazon SNS per il monitoraggio CloudTrail, la sicurezza e le notifiche.
- Patch e aggiornamenti software completamente gestiti.
- AWSIntegrazione di Identity and Access Management (IAM) e controllo degli accessi basato su tag per le API di gestione.

Componenti core MemoryDB

Di seguito è disponibile una panoramica dei componenti principali di una distribuzione MemoryDB.

Argomenti

- [Cluster](#)
- [Nodi](#)
- [Shard](#)
- [Gruppi di parametri](#)

- [Gruppi di sottoreti](#)
- [Liste di controllo accessi](#)
- [Utenti](#)

Cluster

Un cluster è una raccolta di uno o più nodi che servono un singolo set di dati. Un set di dati MemoryDB è partizionato in shard e ogni shard dispone di un nodo primario e un massimo di 5 nodi di replica opzionali. Un nodo principale serve richieste di lettura e scrittura, mentre una replica serve solo richieste di lettura. Un nodo primario può eseguire il failover su un nodo di replica, promuovendo la replica nel nuovo nodo primario per tale shard. MemoryDB esegue Redis come motore di database e, quando si crea un cluster, si specifica la versione di Redis per il cluster. Puoi creare e modificare un cluster utilizzando laAWS CLI, l'API MemoryDB o ilAWS Management Console.

Ogni cluster MemoryDB esegue una versione del motore di Redis. Ogni versione del motore Redis ha le proprie funzionalità supportate. Inoltre, ogni versione del motore Redis dispone di un set di parametri in un gruppo di parametri in grado di controllare il comportamento dei cluster che gestisce.

La capacità di calcolo e memoria di un cluster è determinata dal tipo di nodo. Puoi selezionare il tipo di nodo più adatto alle tue esigenze. Se le tue esigenze cambiano nel tempo, potrai cambiare i tipi di nodo. Per informazioni, consultare [Tipi di nodi supportati](#).

Note

Per informazioni sui prezzi dei tipi di nodi MemoryDB, consulta [Prezzi di MemoryDB](#).

Puoi eseguire un cluster su un VPC tramite il servizio Amazon Virtual Private Cloud (Amazon VPC). Quando utilizzi un VPC, hai il controllo completo sull'ambiente virtuale di rete. Puoi scegliere il tuo intervallo di indirizzi IP, creare sottoreti e configurare liste di routing e di controllo accessi. MemoryDB gestisce snapshot, patch software, rilevamento automatico dei guasti e ripristino. Non è previsto alcun costo aggiuntivo per eseguire il cluster in un VPC. Per ulteriori informazioni sull'utilizzo di Amazon VPC con MemoryDB, consulta [MemoryDB e Amazon VPC](#).

Molte operazioni MemoryDB sono dedicate ai cluster:

- Creazione di un cluster
- Modifica di un cluster

- Esecuzione di snapshot di un cluster
- Eliminazione di un cluster
- Visualizzazione degli elementi in un cluster
- Aggiunta o rimozione di tag di allocazione costi a e da un cluster

Per informazioni più dettagliate, consulta i seguenti argomenti correlati:

- [Gestione dei cluster](#) e [Gestione di nodi](#)

Informazioni su cluster, nodi e operazioni correlate.

- [Resilienza in MemoryDB per Redis](#)

Informazioni su come migliorare la tolleranza ai guasti dei cluster.

Nodi

Un nodo è il più piccolo elemento di base di una distribuzione MemoryDB ed viene eseguito utilizzando un'istanza Amazon EC2. Ogni nodo esegue la versione di Redis scelta al momento della creazione del cluster. Un nodo appartiene a uno shard che appartiene a un cluster.

Ogni nodo esegue un'istanza del motore nella versione scelta al momento della creazione del cluster. Se necessario, puoi eseguire il dimensionamento verso l'alto o verso il basso dei nodi di un cluster in un tipo diverso. Per ulteriori informazioni, consultare [Dimensionamento](#).

Ogni nodo all'interno di un cluster è dello stesso tipo di nodo. Sono supportati più tipi di nodi, ciascuno con quantità variabili di memoria. Per un elenco dei tipi di nodo supportati, consulta [Tipi di nodi supportati](#).

Per ulteriori informazioni sui nodi, consulta [Gestione di nodi](#).

Shard

Uno shard è un raggruppamento da uno a 6 nodi, uno che funge da nodo di scrittura principale e gli altri 5 che fungono da repliche di lettura. Un cluster MemoryDB ha sempre almeno uno shard.

I cluster MemoryDB possono contenere fino a 500 frammenti, con i dati partizionati tra i frammenti. Ad esempio, è possibile scegliere di configurare un cluster a 500 nodi che varia tra 83 shard (un primario e 5 repliche per partizione) e 500 shard (un singolo primario e nessuna replica). Assicurati

che esistano abbastanza indirizzi IP disponibili per soddisfare l'aumento. Le problematiche comuni sono che le sottoreti nel gruppo di sottoreti hanno un intervallo CIDR troppo piccolo o che le sottoreti sono condivise e utilizzate pesantemente da altri cluster.

Una partizione a nodo multiplo implementa repliche tramite un nodo primario di lettura/scrittura e 1-5 nodi di replica. Per ulteriori informazioni, consultare [Informazioni sulla replica MemoryDB](#).

Per ulteriori informazioni sugli shard, consulta [Utilizzo degli shard](#).

Gruppi di parametri

I gruppi di parametri sono un modo facile per gestire le impostazioni di runtime per Redis sul cluster. I parametri vengono utilizzati per controllare l'utilizzo della memoria, le dimensioni degli elementi e altro ancora. Un gruppo di parametri MemoryDB è una raccolta con nome di parametri specifici per motore applicabili a un cluster e tutti i nodi di quel cluster sono configurati esattamente nello stesso modo.

Per informazioni più dettagliate sui gruppi di parametri MemoryDB, consulta [Configurazione dei parametri di motore con i gruppi di parametri](#).

Gruppi di sottoreti

Un gruppo di sottoreti è una raccolta di sottoreti (generalmente private) che è possibile designare per i cluster in esecuzione in un ambiente Amazon Virtual Private Cloud (VPC)

Quando crei un cluster in un Amazon VPC, puoi specificare un gruppo di sottoreti o utilizzare quello predefinito fornito. MemoryDB usa il gruppo di sottoreti per selezionare una sottorete e un indirizzo IP all'interno di quella sottorete da associare ai nodi.

Per informazioni più dettagliate sui gruppi di sottoreti MemoryDB, consulta [Sottoreti e gruppi di sottoreti](#).

Liste di controllo accessi

Un elenco di controllo di accesso è una raccolta di uno o più utenti. Le stringhe di accesso seguono Redis [Regole di controllo accessi](#) per autorizzare l'accesso dell'utente ai comandi e ai dati di Redis.

Per informazioni più dettagliate sugli elenchi di controllo accessi MemoryDB, vedere [Autenticazione degli utenti con elenchi di controllo degli accessi \(ACL\)](#).

Utenti

Un utente ha un nome utente e una password e viene utilizzato per accedere ai dati e inviare comandi sul cluster MemoryDB. Un utente è membro di un Access Control List (ACL), che è possibile utilizzare per determinare le autorizzazioni per quell'utente sui cluster MemoryDB. Per ulteriori informazioni, consulta [Autenticazione degli utenti con elenchi di controllo degli accessi \(ACL\)](#)

Servizi correlati

[ElastiCache per Redis](#)

Nel decidere se utilizzare MemoryDB per Redis o ElastiCache per Redis, considera i seguenti confronti:

- MemoryDB for Redis è un database durevole in memoria per carichi di lavoro che richiedono un database primario ultraveloce. Occorre considerare l'utilizzo di MemoryDB se il carico di lavoro richiede un database durevole che offra prestazioni ultraveloci (latenza di lettura di microsecondi e latenza di scrittura di millisecondi a una cifra). MemoryDB può anche essere adatto al caso d'uso se si desidera creare un'applicazione utilizzando strutture di dati e API Redis con un database primario, durevole. Infine, occorre considerare l'utilizzo di MemoryDB per semplificare l'architettura dell'applicazione e ridurre i costi sostituendo l'utilizzo di un database con una cache per garantire durata e prestazioni.
- ElastiCache for Redis è un servizio che viene comunemente utilizzato per memorizzare nella cache i dati da altri database e archivi dati utilizzando Redis. È consigliabile utilizzare Redis ElastiCache per la memorizzazione nella cache dei carichi di lavoro in cui si desidera accelerare l'accesso ai dati con il database o l'archivio dati primario esistente (prestazioni di lettura e scrittura in microsecondi). Dovresti anche prendere in considerazione Redis ElastiCache per i casi d'uso in cui desideri utilizzare le strutture di dati e le API Redis per accedere ai dati archiviati in un database o in un data store primario.

Scelta di regioni e zone di disponibilità

Le risorse di cloud computing di AWS si trovano in strutture di data center ad alta disponibilità. Per fornire ulteriore scalabilità e affidabilità, queste strutture di data center sono in ubicazioni fisiche diverse. Tali località sono categorizzate in base a regioni e zone di disponibilità.

Le regioni AWS sono di grandi dimensioni e ampiamente distribuite in aree geografiche separate. Le zone di disponibilità sono percorsi separati all'interno di una regione AWS, costruiti per essere isolati dai potenziali errori di altre zone di disponibilità. Offrono una connettività di rete non costosa e a bassa latenza ad altre zone di disponibilità nella stessa regione AWS.

 Important

Ciascuna regione è completamente indipendente. Qualsiasi attività di MemoryDB avviata (ad esempio, la creazione di cluster) viene eseguita solo nella regione predefinita corrente.

Per creare o utilizzare un cluster in una regione specifica, utilizza l'endpoint del servizio regionale corrispondente. Per gli endpoint del servizio, consulta [Regioni ed endpoint supportati](#).

Ubicazione dei nodi

Qualsiasi cluster con almeno una replica deve essere distribuito tra AZ. L'unico modo per localizzare tutto all'interno di una singola AZ è con un cluster composto da shard a nodo singolo.

Localizzando i nodi in diverse AZ, MemoryDB elimina la possibilità che un guasto, ad esempio un'interruzione dell'alimentazione, in una zona AZ provochi una perdita di disponibilità.

- [Creazione di un cluster MemoryDB](#)
- [Modifica di un cluster MemoryDB](#)

Regioni ed endpoint supportati

MemoryDB per Redis è disponibile in più regioni. AWS Ciò significa che puoi avviare cluster MemoryDB in posizioni che soddisfano i tuoi requisiti. Ad esempio, puoi avviarlo nella regione AWS più vicina ai clienti o in una regione AWS specifica per soddisfare determinati requisiti legali. Inoltre, man mano che MemoryDB espande la disponibilità in una nuova AWS regione, MemoryDB supporta le due MAJOR.MINOR versioni più recenti dell'epoca per la nuova regione. Per ulteriori informazioni sulle versioni di MemoryDB, vedere. [Versioni del motore Redis](#)

Per impostazione predefinita, gli AWS SDK, l'API MemoryDB e la console MemoryDB fanno riferimento alla regione Stati Uniti orientali (Virginia settentrionale). AWS CLI Man mano che MemoryDB estende la disponibilità a nuove regioni, sono disponibili anche nuovi endpoint per queste aree da utilizzare nelle richieste HTTP, negli SDK e nella console. AWS AWS CLI

Ogni regione è pensata per essere completamente isolata dalle altre regioni . All'interno di ciascuna regione ci sono più zone di disponibilità. Avviando i nodi in diverse AZ si ottiene la massima tolleranza ai guasti possibile. Per ulteriori informazioni su regioni e zone di disponibilità, vedere [Scelta di regioni e zone di disponibilità](#) all'inizio di questo argomento.

Regioni in cui è supportato MemoryDB

Nome regione/Regione	Endpoint	Protocollo	
Stati Uniti orientali (Ohio) us-east-2	memory-db.us-east-2.amazonaws.com	HTTPS	
Stati Uniti orientali (Virginia settentrionale) us-east-1	memory-db.us-east-1.amazonaws.com	HTTPS	
Regione Stati Uniti occidentali (California settentrionale) us-west-1	memory-db.us-west-1.amazonaws.com	HTTPS	

Nome regione/Regione	Endpoint	Protocollo
Stati Uniti occidentali (Oregon) us-west-2	memory-db.us-west-2.amazonaws.com	HTTPS
Regione Canada (Centrale) ca-central-1	memory-db.ca-central-1.amazonaws.com	HTTPS
Regione Asia Pacifico (Hong Kong) ap-east-1	memory-db.ap-east-1.amazonaws.com	HTTPS
Regione Asia Pacifico (Mumbai) ap-south-1	memory-db.ap-south-1.amazonaws.com	HTTPS
Regione Asia Pacifico (Tokyo) ap-northeast-1	memory-db.ap-northeast-1.amazonaws.com	HTTPS
Regione Asia Pacifico (Seoul) ap-northeast-2	memory-db.ap-northeast-2.amazonaws.com	HTTPS
Regione Asia Pacifico (Singapore) ap-southeast-1	memory-db.ap-southeast-1.amazonaws.com	HTTPS

Nome regione/Regione	Endpoint	Protocollo	
Asia Pacifico (Sydney) ap-southeast-2	memory-db.ap-southeast-2.amazonaws.com	HTTPS	
Regione Europa (Francoforte) eu-central-1	memory-db.eu-central-1.amazonaws.com	HTTPS	
Europa (Irlanda) eu-west-1	memory-db.eu-west-1.amazonaws.com	HTTPS	
Regione Europa (Londra) eu-west-2	memory-db.eu-west-2.amazonaws.com	HTTPS	
Regione UE (Parigi) eu-west-3	memory-db.eu-west-3.amazonaws.com	HTTPS	
Regione Europa (Stoccolma) eu-north-1	memory-db.eu-north-1.amazonaws.com	HTTPS	
Regione Europa (Milano) eu-south-1	memory-db.eu-south-1.amazonaws.com	HTTPS	
Regione Sud America (San Paolo) sa-east-1	memory-db.sa-east-1.amazonaws.com	HTTPS	

Nome regione/Regione	Endpoint	Protocollo
Regione Cina (Pechino) cn-north-1	memory-db.cn-north-1.amazonaws.com.cn	HTTPS
Regione Cina (Ningxia) cn-northwest-1	memory-db.cn-northwest-1.amazonaws.com.cn	HTTPS

Per una tabella di AWS prodotti e servizi per regione, vedi [Prodotti e servizi per regione](#).

Per una tabella delle zone di disponibilità supportate all'interno delle regioni, vedi [Sottoreti e gruppi di sottoreti](#).

Accesso a MemoryDB

Ogni endpoint del cluster MemoryDB contiene un indirizzo e una porta. Questo endpoint del cluster supporta il protocollo Redis Cluster per consentire ai client di scoprire ruoli, indirizzi IP e slot specifici per ogni nodo del cluster. Quando un nodo primario si guasta e al suo posto viene promossa una replica, puoi connetterti all'endpoint del cluster per scoprire il nuovo nodo primario utilizzando il protocollo Redis Cluster.

È necessario connettersi all'endpoint del cluster per scoprire gli endpoint del nodo utilizzando il comando `cluster nodes cluster slots`. Dopo aver scoperto il nodo giusto per una chiave, puoi connetterti direttamente al nodo per le richieste di lettura/scrittura. Un client Redis può utilizzare l'endpoint del cluster per connettersi automaticamente al nodo corretto.

Per risolvere i problemi di nodi specifici in un cluster, puoi anche utilizzare endpoint specifici del nodo, ma questi non sono necessari per il normale utilizzo.

Per trovare l'endpoint di un cluster, consulta quanto segue:

- [Ricerca dell'endpoint per un cluster MemoryDB \(AWSCLIP\)](#)
- [Ricerca dell'endpoint per un cluster MemoryDB \(API MemoryDB\)](#)

Per la connessione a nodi o cluster, vedi. [Connessione ai nodi MemoryDB utilizzando redis-cli](#)

Sicurezza di MemoryDB

La sicurezza per MemoryDB è gestita a tre livelli:

- Per controllare chi può eseguire azioni di gestione su cluster e nodi di MemoryDB, si utilizza AWS Identity and Access Management (IAM). Quando ti connetti AWS utilizzando le credenziali IAM, il tuo AWS account deve disporre di politiche IAM che concedano le autorizzazioni necessarie per eseguire le operazioni. Per ulteriori informazioni, consultare [Gestione delle identità e degli accessi in MemoryDB per Redis](#)
- Per controllare i livelli di accesso ai cluster, crei utenti con autorizzazioni specifiche e li assegni agli Access Control List (ACL). L'ACL, a sua volta, viene quindi associato a uno o più cluster. Per ulteriori informazioni, consulta [Autenticazione degli utenti con elenchi di controllo degli accessi \(ACL\)](#).
- I cluster MemoryDB devono essere creati in un cloud privato virtuale (VPC) basato sul servizio Amazon VPC. Per controllare quali dispositivi e istanze Amazon EC2 possono aprire connessioni all'endpoint e alla porta del nodo per i cluster MemoryDB in un VPC, utilizzi un gruppo di sicurezza VPC. Puoi creare queste connessioni di endpoint e porta tramite Transport Layer Security (TLS)/ Secure Sockets Layer (SSL). Inoltre, le regole del firewall della tua azienda possono controllare se i dispositivi in esecuzione nella tua azienda possono aprire connessioni a un cluster MemoryDB. Per ulteriori informazioni sui VPC, consulta [MemoryDB e Amazon VPC](#).

Per informazioni sulla configurazione della sicurezza, vedi [Sicurezza in MemoryDB for Redis](#).

Guida introduttiva a MemoryDB

Questo esercizio illustra i passaggi per creare, concedere l'accesso, connettersi e infine eliminare un cluster MemoryDB utilizzando la console di gestione di MemoryDB.

Note

Ai fini di questo esercizio, si consiglia di utilizzare l'opzione Easy create durante la creazione di un cluster e di tornare alle altre due opzioni dopo aver esplorato ulteriormente le funzionalità di MemoryDB.

Argomenti

- [Configurazione](#)
- [Fase 1: creazione di un cluster](#)
- [Passaggio 2: autorizzare l'accesso al cluster](#)
- [Fase 3: Connect al cluster](#)
- [Fase 4: Eliminazione di un cluster](#)
- [A questo punto come si può procedere?](#)

Configurazione

Di seguito, puoi trovare argomenti che descrivono le azioni una tantum da intraprendere per iniziare a utilizzare MemoryDB.

Argomenti

- [Crea il tuo account AWS](#)
- [Concessione dell'accesso programmatico](#)
- [Configura le tue autorizzazioni \(solo nuovi utenti di MemoryDB\)](#)
- [Scaricamento e configurazione della CLI AWS](#)

Crea il tuo account AWS

Registrati per un Account AWS

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

Per iscriverti a un Account AWS

1. Apri la pagina all'indirizzo <https://portal.aws.amazon.com/billing/signup>.
2. Segui le istruzioni online.

Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

Quando ti iscrivi a un Account AWS, Utente root dell'account AWS viene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come procedura consigliata in materia di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire [attività che richiedono l'accesso da parte dell'utente root](#).

AWS ti invia un'e-mail di conferma dopo il completamento della procedura di registrazione. È possibile visualizzare l'attività corrente dell'account e gestire l'account in qualsiasi momento accedendo all'indirizzo <https://aws.amazon.com/> e selezionando Il mio account.

Crea un utente con accesso amministrativo

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

Proteggi i tuoi Utente root dell'account AWS

1. Accedi [AWS Management Console](#) come proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

Per informazioni sull'accesso utilizzando un utente root, consulta la pagina [Signing in as the root user](#) della Guida per l'utente di Accedi ad AWS .

2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

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

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

Per istruzioni, consulta [Abilitazione di AWS IAM Identity Center](#) nella Guida per l'utente di AWS IAM Identity Center .

2. In IAM Identity Center, concedi l'accesso amministrativo a un utente.

Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta [Configurare l'accesso utente con le impostazioni predefinite IAM Identity Center directory](#) nella Guida per l'AWS IAM Identity Center utente.

Accedi come utente con accesso amministrativo

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

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

Assegna l'accesso ad altri utenti

1. In IAM Identity Center, crea un set di autorizzazioni che segua la migliore pratica di applicazione delle autorizzazioni con privilegi minimi.

Per istruzioni, consulta [Creare un set di autorizzazioni](#) nella Guida per l'utente.AWS IAM Identity Center

2. Assegna gli utenti a un gruppo, quindi assegna l'accesso Single Sign-On al gruppo.

Per istruzioni, consulta [Aggiungere gruppi](#) nella Guida per l'utente.AWS IAM Identity Center

Concessione dell'accesso programmatico

Gli utenti hanno bisogno di un accesso programmatico se vogliono interagire con l' AWS AWS Management Console esterno di. Il modo per concedere l'accesso programmatico dipende dal tipo di utente che accede. AWS

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

Quale utente necessita dell'accesso programmatico?	Per	Come
Identità della forza lavoro (Utenti gestiti nel centro identità IAM)	Utilizza credenziali temporane e per firmare le richieste programmatiche agli AWS CLI AWS SDK o alle API. AWS	<p>Segui le istruzioni per l'interfaccia che desideri utilizzare.</p> <ul style="list-style-type: none"> • Per la AWS CLI, consulta Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'utente.AWS Command Line Interface • Per AWS SDK, strumenti e AWS API, consulta l'autenticazione IAM Identity Center nella Guida di riferimento agli AWS SDK e agli strumenti.
IAM	Utilizza credenziali temporane e per firmare le richieste programmatiche agli SDK o alle API AWS CLI. AWS AWS	Segui le istruzioni in Uso delle credenziali temporanee con AWS risorse nella Guida per l'utente IAM.
IAM	(Non consigliato) Utilizza credenziali a lungo termine per firmare le richieste programmatiche agli AWS CLI AWS SDK o alle API. AWS	<p>Segui le istruzioni per l'interfaccia che desideri utilizzare.</p> <ul style="list-style-type: none"> • Per la AWS CLI, consulta Autenticazione tramite credenziali utente IAM nella Guida per l'utente.AWS Command Line Interface • Per gli AWS SDK e gli strumenti, consulta Autenticazione tramite credenziali a lungo termine nella Guida di riferimen

Quale utente necessita dell'accesso programmatico?	Per	Come
		<p>to agli SDK e agli AWS strumenti.</p> <ul style="list-style-type: none"> • Per le AWS API, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente IAM.

Argomenti correlati:

- [Cos'è IAM?](#) nella Guida per l'utente di IAM.
- [AWS Informazioni generali sulle credenziali di sicurezza](#).AWS

Configura le tue autorizzazioni (solo nuovi utenti di MemoryDB)

Per fornire l'accesso, aggiungi autorizzazioni ai tuoi utenti, gruppi o ruoli:

- Utenti e gruppi in: AWS IAM Identity Center

Crea un set di autorizzazioni. Segui le istruzioni riportate nella pagina [Create a permission set](#) (Creazione di un set di autorizzazioni) nella Guida per l'utente di AWS IAM Identity Center .

- Utenti gestiti in IAM tramite un provider di identità:

Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina [Creating a role for a third-party identity provider \(federation\)](#) (Creazione di un ruolo per un provider di identità di terze parti [federazione]) nella Guida per l'utente di IAM.

- Utenti IAM:
 - Crea un ruolo che l'utente possa assumere. Per istruzioni, consulta la pagina [Creating a role for an IAM user](#) (Creazione di un ruolo per un utente IAM) nella Guida per l'utente di IAM.
 - (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate nella pagina [Aggiunta di autorizzazioni a un utente \(console\)](#) nella Guida per l'utente di IAM.

MemoryDB for Redis crea e utilizza ruoli collegati ai servizi per fornire risorse e accedere ad altre AWS risorse e servizi per conto dell'utente. Per consentire a MemoryDB di creare per voi un ruolo collegato ai servizi, utilizzate la policy -managed denominata `AWSAmazonMemoryDBFullAccess`. Per questo ruolo viene effettuato il provisioning preventivo con l'autorizzazione necessaria al servizio per creare un ruolo collegato ai servizi per tuo conto.

Potresti decidere di non utilizzare la policy predefinita e di utilizzare piuttosto una policy gestita in modo personalizzato. In questo caso, assicurati di disporre delle autorizzazioni per la chiamata `iam:createServiceLinkedRole` o di aver creato il ruolo collegato al servizio MemoryDB.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Creare una nuova policy \(IAM\)](#)
- [AWSPolicy gestite da \(predefinite\) per MemoryDB for Redis](#)
- [Utilizzo di ruoli collegati ai servizi per Amazon MemoryDB for Redis](#)

Scaricamento e configurazione della CLI AWS

AWS CLI [È disponibile all'indirizzo http://aws.amazon.com/cli](http://aws.amazon.com/cli). Viene eseguita in Windows, MacOS e Linux. Dopo averlo scaricato AWS CLI, segui questi passaggi per installarlo e configurarlo:

1. Consulta la [Guida per l'utente di AWS per l'interfaccia a riga di comando](#)
2. Segui le istruzioni per l'[installazione della AWS CLI](#) e la [configurazione della CLI](#). AWS

Fase 1: creazione di un cluster

Prima di creare un cluster per l'utilizzo in produzione, è ovviamente necessario considerare come configurare il cluster per soddisfare le esigenze aziendali. Tali problemi sono affrontati nella sezione [Preparazione di un cluster](#). Ai fini di questo esercizio introduttivo, è possibile accettare i valori di configurazione predefiniti laddove applicabili.

Il cluster che stai per avviare verrà eseguito in un ambiente attivo e non in una sandbox. Fino a quando non la elimini, ti verranno addebitati i costi di utilizzo standard di MemoryDB per l'istanza. L'addebito totale sarà minimo (in genere meno di un dollaro) se completi l'esercizio descritto qui in una sola seduta ed elimini il cluster alla fine. [Per ulteriori informazioni sui tassi di utilizzo di MemoryDB, consulta MemoryDB.](#)

Il cluster viene avviato in un virtual private cloud (VPC) basato sul servizio Amazon VPC.

Creazione di un cluster MemoryDB

I seguenti esempi mostrano come creare un cluster utilizzando l'API AWS CLI e AWS Management Console MemoryDB.

Creazione di un cluster (Console)

Per creare un cluster utilizzando la console MemoryDB

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Scegli Clusters nel riquadro di navigazione a sinistra, quindi scegli Crea.

Easy create

1. Completare la sezione Configurazione. Questo configura il tipo di nodo e la configurazione predefinita del cluster. Seleziona la dimensione di memoria e le prestazioni di rete appropriate tra le seguenti opzioni:
 - Produzione
 - Sviluppo/Test
 - Demo
2. Completa la sezione Informazioni sul cluster.

- a. Nel campo Name (Nome), immettere un nome per il cluster.

I vincoli di denominazione dei cluster sono i seguenti:

- Devono contenere da 1 a 40 caratteri alfanumerici o trattini.
- Devono iniziare con una lettera.
- Non possono contenere due trattini consecutivi.
- Non possono terminare con un trattino.

- b. Nella casella Description (Descrizione), immettere una descrizione per questo cluster

3. Completa la sezione Gruppi di sottoreti:

- Per i gruppi di sottoreti, crea un nuovo gruppo di sottoreti o scegline uno esistente dall'elenco disponibile che desideri applicare a questo cluster. Se ne stai creando uno nuovo:
 - Inserisci un nome
 - Inserisci una descrizione
 - Se è stata attivata la funzione Multi-AZ, il gruppo di sottoreti deve contenere almeno due sottoreti che risiedono in zone di disponibilità diverse. Per ulteriori informazioni, consulta [Sottoreti e gruppi di sottoreti](#).
 - Se stai creando un nuovo gruppo di sottoreti e non disponi di un VPC esistente, ti verrà chiesto di creare un VPC. Per ulteriori informazioni, consultare [Che cos'è Amazon VPC?](#) nella Guida per l'utente di Amazon VPC

4. Per la ricerca vettoriale, puoi abilitare la funzionalità di ricerca vettoriale per memorizzare incorporamenti vettoriali ed eseguire ricerche vettoriali. Nota che questo correggerà i valori per la compatibilità della versione Redis, i gruppi di parametri e gli Shards. Per ulteriori informazioni, consulta [Ricerca vettoriale](#).

5. Visualizza le impostazioni predefinite:

Quando si utilizza Easy create, le impostazioni del cluster rimanenti sono impostate di default. Nota che alcune di queste impostazioni possono essere modificate dopo la creazione, come indicato da Editable dopo la creazione.

6. Per i tag, puoi facoltativamente applicare tag per cercare e filtrare i cluster o tenere traccia dei costi. AWS

7. Riesaminare le voci e le selezioni, quindi apportare le eventuali correzioni. Quando sei pronto, scegli Crea per avviare il cluster o Annulla per annullare l'operazione.

Quando lo stato del cluster diventa disponibile, è possibile concedere a EC2 l'accesso al cluster, connettersi al cluster e iniziare a utilizzarlo. Per ulteriori informazioni, consultare [Passaggio 2: autorizzare l'accesso al cluster](#)

⚠ Important

Non appena il cluster diventa disponibile, viene addebitata ogni ora o frazione di ora in cui il cluster è attivo, anche se non viene effettivamente utilizzato. Per evitare di sostenere i costi del cluster, è necessario eliminarlo. Per informazioni, consulta [Fase 4: Eliminazione di un cluster](#).

Create new cluster

1. Completa la sezione Informazioni sul cluster.
 - a. Nel campo Name (Nome), immettere un nome per il cluster.

I vincoli di denominazione dei cluster sono i seguenti:

 - Devono contenere da 1 a 40 caratteri alfanumerici o trattini.
 - Devono iniziare con una lettera.
 - Non possono contenere due trattini consecutivi.
 - Non possono terminare con un trattino.
 - b. Nella casella Description (Descrizione), immettere una descrizione per questo cluster
2. Completa la sezione Gruppi di sottoreti:
 - Per i gruppi di sottoreti, crea un nuovo gruppo di sottoreti o scegline uno esistente dall'elenco disponibile che desideri applicare a questo cluster. Se ne stai creando uno nuovo:
 - Inserisci un nome
 - Inserisci una descrizione

- Se è stata attivata la funzione Multi-AZ, il gruppo di sottoreti deve contenere almeno due sottoreti che risiedono in zone di disponibilità diverse. Per ulteriori informazioni, consulta [Sottoreti e gruppi di sottoreti](#).
- Se stai creando un nuovo gruppo di sottoreti e non disponi di un VPC esistente, ti verrà chiesto di creare un VPC. Per ulteriori informazioni, consultare [Che cos'è Amazon VPC?](#) nella Guida per l'utente di Amazon VPC

3. Completa la sezione Impostazioni del cluster:

- a. Per abilitare la funzionalità di ricerca vettoriale, è possibile abilitarla per memorizzare incorporamenti vettoriali ed eseguire ricerche vettoriali. Nota che questo correggerà i valori per la compatibilità della versione Redis, i gruppi di parametri e gli Shards. Per ulteriori informazioni, consulta [Ricerca vettoriale](#).
- b. Per la compatibilità della versione Redis, accetta l'impostazione predefinita. 6.2
- c. Per Port, accetta la porta Redis predefinita 6379 o, se hai un motivo per utilizzare una porta diversa, inserisci il numero di porta..
- d. Per il gruppo di parametri, se hai abilitato la ricerca vettoriale, usa. `default.memorydb-redis7.search.preview` Altrimenti, accettate il gruppo di `default.memorydb-redis7` parametri.

I gruppi di parametri controllano i parametri di runtime del cluster. Per ulteriori informazioni sui gruppi di parametri, consulta [Parametri specifici di Redis](#).

- e. Per Tipo di nodo, scegliete un valore per il tipo di nodo (insieme alla dimensione della memoria associata) che desiderate.

Se si sceglie un tipo di nodo dalla famiglia r6gd, si abilita automaticamente il tiering di dati, che divide l'archiviazione dati tra memoria e SSD. Per ulteriori informazioni, consulta [Tiering di dati](#).

- f. Per Numero di shard, scegliete il numero di shard che desiderate per questo cluster. Per una maggiore disponibilità dei cluster, ti consigliamo di aggiungere almeno 2 shard.

È possibile modificare il numero di shard nel cluster in modo dinamico. Per ulteriori informazioni, consulta [Dimensionamento di cluster di MemoryDB](#).


- g. In Replicas per shard (Repliche per partizione): scegliere il numero di nodi di replica di lettura per ogni partizione.

Esistono le seguenti restrizioni:

- Se hai abilitato la funzione Multi-AZ, assicurati di avere almeno una replica per ogni partizioni.
 - Quando utilizzi la console per creare il cluster, il numero delle repliche è lo stesso per ogni partizioni.
- h. Seleziona Next (Successivo).
- i. Completa la sezione Impostazioni avanzate:
- i. In Security groups (Gruppi di sicurezza), scegliere i gruppi di sicurezza per il cluster. Un gruppo di sicurezza si comporta come un firewall, controllando l'accesso di rete al cluster. È possibile utilizzare il gruppo di sicurezza di default per il VPC o crearne uno nuovo.

Per ulteriori informazioni sui gruppi di sicurezza, consulta [Gruppi di sicurezza per il VPC](#) nella Guida per l'utente di Amazon VPC.

- ii. Per crittografare i dati, le opzioni disponibili sono le seguenti:
- Crittografia dei dati inattivi : Consente la crittografia dei dati memorizzati su disco. Per ulteriori informazioni, consultare [Crittografia dei dati inattivi](#).

 Note

Hai la possibilità di fornire una chiave di crittografia diversa da quella predefinita scegliendo la chiave KMS di AWS proprietà di Customer Managed e scegliendo la chiave.

- Crittografia dei dati in transito : Consente la crittografia dei dati in trasferimento. Se non si seleziona alcuna crittografia, verrà creato un elenco di controllo degli accessi aperto denominato «accesso aperto» con un utente predefinito. Per ulteriori informazioni, consulta [Autenticazione degli utenti con elenchi di controllo degli accessi \(ACL\)](#).
- iii. Per Snapshot, specificate facoltativamente un periodo di conservazione delle istantanee e una finestra per le istantanee. Per impostazione predefinita, l'opzione Abilita istantanee automatiche è preselezionata.
- iv. Per la finestra di manutenzione, specificare facoltativamente una finestra di manutenzione. La finestra di manutenzione è l'orario, generalmente di un'ora, ogni settimana in cui MemoryDB pianifica la manutenzione del sistema per il cluster.

È possibile consentire a MemoryDB di scegliere il giorno e l'ora per la finestra di manutenzione (nessuna preferenza), oppure è possibile scegliere autonomamente il giorno, l'ora e la durata (Specificare la finestra di manutenzione). Se dagli elenchi si sceglie Specify maintenance window (Specifica finestra di manutenzione), selezionare Start day (Giorno di inizio), Start time (Ora di inizio) e Duration (Durata) (in ore) per la finestra di manutenzione. Tutti gli orari si intendono in formato UCT.

Per ulteriori informazioni, consulta [Gestione della manutenzione](#).

- v. In Notifications (Notifiche), scegliere un argomento esistente di Amazon Simple Notification Service (Amazon SNS) o scegliere l'input manuale dell'ARN nell'Amazon Resource Name (ARN) dell'argomento. Amazon SNS permette di inviare notifiche ai dispositivi intelligenti connessi a Internet. Le notifiche sono disabilitate per impostazione predefinita. [Per ulteriori informazioni, consulta https://aws.amazon.com/sns/](https://aws.amazon.com/sns/).
- vi. Per i tag, puoi facoltativamente applicare tag per cercare e filtrare i cluster o tenere traccia AWS dei costi.
- j. Riesaminare le voci e le selezioni, quindi apportare le eventuali correzioni. Quando sei pronto, scegli Crea per avviare il cluster o Annulla per annullare l'operazione.

Quando lo stato del cluster diventa disponibile, è possibile concedere a EC2 l'accesso al cluster, connettersi al cluster e iniziare a utilizzarlo. Per ulteriori informazioni, consultare [Passaggio 2: autorizzare l'accesso al cluster](#)

Important

Non appena il cluster diventa disponibile, viene addebitata ogni ora o frazione di ora in cui il cluster è attivo, anche se non viene effettivamente utilizzato. Per evitare di sostenere i costi del cluster, è necessario eliminarlo. Per informazioni, consulta [Fase 4: Eliminazione di un cluster](#).

Restore from snapshots

In Origine istantanea, scegli l'istantanea di origine da cui migrare i dati. Per ulteriori informazioni, consulta [Snapshot e ripristino](#).

Note

Se desideri che nel nuovo cluster sia abilitata la ricerca vettoriale, anche lo snapshot di origine deve avere abilitata la ricerca vettoriale.

Per impostazione predefinita, il cluster di destinazione utilizza le impostazioni del cluster di origine. Facoltativamente, è possibile modificare le seguenti impostazioni sul cluster di destinazione:

1. Informazioni sul cluster

- a. Nel campo Name (Nome), immettere un nome per il cluster.

I vincoli di denominazione dei cluster sono i seguenti:

- Devono contenere da 1 a 40 caratteri alfanumerici o trattini.
- Devono iniziare con una lettera.
- Non possono contenere due trattini consecutivi.
- Non possono terminare con un trattino.

- b. Nella casella Description (Descrizione), immettere una descrizione per questo cluster

2. Gruppi di sottoreti

- Per i gruppi di sottoreti, crea un nuovo gruppo di sottoreti o scegline uno esistente dall'elenco disponibile che desideri applicare a questo cluster. Se ne stai creando uno nuovo:
 - Inserisci un nome
 - Inserisci una descrizione
 - Se è stata attivata la funzione Multi-AZ, il gruppo di sottoreti deve contenere almeno due sottoreti che risiedono in zone di disponibilità diverse. Per ulteriori informazioni, consulta [Sottoreti e gruppi di sottoreti](#).
 - Se stai creando un nuovo gruppo di sottoreti e non disponi di un VPC esistente, ti verrà chiesto di creare un VPC. Per ulteriori informazioni, consultare [Che cos'è Amazon VPC?](#) nella Guida per l'utente di Amazon VPC

3. Impostazioni del cluster

- a. Per abilitare la funzionalità di ricerca vettoriale, è possibile abilitarla per memorizzare incorporamenti vettoriali ed eseguire ricerche vettoriali. Nota che questo correggerà i valori per la compatibilità della versione Redis, i gruppi di parametri e gli Shards. Per ulteriori informazioni, consulta [Ricerca vettoriale](#).
- b. Per la compatibilità della versione Redis, accetta l'impostazione predefinita. 6.2
- c. Per Port, accetta la porta Redis predefinita 6379 o, se hai un motivo per utilizzare una porta diversa, inserisci il numero di porta..
- d. Per il gruppo di parametri, se hai abilitato la ricerca vettoriale, usa. `default.memorydb-redis7.search.preview` Altrimenti, accettate il gruppo di `default.memorydb-redis7` parametri.

I gruppi di parametri controllano i parametri di runtime del cluster. Per ulteriori informazioni sui gruppi di parametri, consulta [Parametri specifici di Redis](#).

- e. Per Tipo di nodo, scegliete un valore per il tipo di nodo (insieme alla dimensione della memoria associata) che desiderate.

Se si sceglie un tipo di nodo dalla famiglia r6gd, si abilita automaticamente il tiering di dati, che divide l'archiviazione dati tra memoria e SSD. Per ulteriori informazioni, consulta [Tiering di dati](#).

- f. Per Numero di shard, scegliete il numero di shard che desiderate per questo cluster. Per una maggiore disponibilità dei cluster, ti consigliamo di aggiungere almeno 2 shard.

È possibile modificare il numero di shard nel cluster in modo dinamico. Per ulteriori informazioni, consulta [Dimensionamento di cluster di MemoryDB](#).

- g. In Replicas per shard (Repliche per partizione): scegliere il numero di nodi di replica di lettura per ogni partizione.


Esistono le seguenti restrizioni:

- Se hai abilitato la funzione Multi-AZ, assicurati di avere almeno una replica per ogni partizione.
 - Quando utilizzi la console per creare il cluster, il numero delle repliche è lo stesso per ogni partizione.
- h. Seleziona Next (Successivo).
 - i. Impostazioni avanzate

- i. In Security groups (Gruppi di sicurezza), scegliere i gruppi di sicurezza per il cluster. Un gruppo di sicurezza si comporta come un firewall, controllando l'accesso di rete al cluster. È possibile utilizzare il gruppo di sicurezza di default per il VPC o crearne uno nuovo.

Per ulteriori informazioni sui gruppi di sicurezza, consulta [Gruppi di sicurezza per il VPC](#) nella Guida per l'utente di Amazon VPC.

- ii. Per crittografare i dati, le opzioni disponibili sono le seguenti:
 - Crittografia dei dati inattivi : Consente la crittografia dei dati memorizzati su disco. Per ulteriori informazioni, consultare [Crittografia dei dati inattivi](#).

 Note

Hai la possibilità di fornire una chiave di crittografia diversa da quella predefinita scegliendo la chiave KMS di AWS proprietà di Customer Managed e scegliendo la chiave.

- Crittografia dei dati in transito : Consente la crittografia dei dati in trasferimento. Se non si seleziona alcuna crittografia, verrà creato un elenco di controllo degli accessi aperto denominato «accesso aperto» con un utente predefinito. Per ulteriori informazioni, consulta [Autenticazione degli utenti con elenchi di controllo degli accessi \(ACL\)](#).
- iii. Per Snapshot, specificate facoltativamente un periodo di conservazione delle istantanee e una finestra per le istantanee. Per impostazione predefinita, l'opzione Abilita istantanee automatiche è preselezionata.
 - iv. Per la finestra di manutenzione, specificare facoltativamente una finestra di manutenzione. La finestra di manutenzione è l'orario, generalmente di un'ora, ogni settimana in cui MemoryDB pianifica la manutenzione del sistema per il cluster. È possibile consentire a MemoryDB di scegliere il giorno e l'ora per la finestra di manutenzione (nessuna preferenza), oppure è possibile scegliere autonomamente il giorno, l'ora e la durata (Specificare la finestra di manutenzione). Se dagli elenchi si sceglie Specify maintenance window (Specifica finestra di manutenzione), selezionare Start day (Giorno di inizio), Start time (Ora di inizio) e Duration (Durata) (in ore) per la finestra di manutenzione. Tutti gli orari si intendono in formato UCT.

Per ulteriori informazioni, consulta [Gestione della manutenzione](#).

- v. In Notifications (Notifiche), scegliere un argomento esistente di Amazon Simple Notification Service (Amazon SNS) o scegliere l'input manuale dell'ARN nell'Amazon Resource Name (ARN) dell'argomento. Amazon SNS permette di inviare notifiche ai dispositivi intelligenti connessi a Internet. Le notifiche sono disabilitate per impostazione predefinita. [Per ulteriori informazioni, consulta https://aws.amazon.com/sns/](https://aws.amazon.com/sns/).
- vi. Per i tag, puoi facoltativamente applicare tag per cercare e filtrare i cluster o tenere traccia AWS dei costi.
- j. Riesaminare le voci e le selezioni, quindi apportare le eventuali correzioni. Quando sei pronto, scegli Crea per avviare il cluster o Annulla per annullare l'operazione.

Quando lo stato del cluster diventa disponibile, è possibile concedere a EC2 l'accesso al cluster, connettersi al cluster e iniziare a utilizzarlo. Per ulteriori informazioni, consultare [Passaggio 2: autorizzare l'accesso al cluster](#)

 Important

Non appena il cluster diventa disponibile, viene addebitata ogni ora o frazione di ora in cui il cluster è attivo, anche se non viene effettivamente utilizzato. Per evitare di sostenere i costi del cluster, è necessario eliminarlo. Per informazioni, consulta [Fase 4: Eliminazione di un cluster](#).

Creazione di un cluster (AWS CLI)

Per creare un cluster utilizzando il AWS CLI, vedere [create-cluster](#). Di seguito è riportato un esempio:

Per 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
```

Per Windows:

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

Dovresti ottenere la seguente risposta 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:xxxxxxxxxxxx:cluster/my-cluster",  
    "SnapshotRetentionLimit": 0,  
    "MaintenanceWindow": "wed:03:00-wed:04:00",
```

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

È possibile iniziare a utilizzare il cluster una volta che il suo stato cambia `inavailable`.

Important

Non appena il cluster diventa disponibile, viene addebitata ogni ora o frazione di ora in cui il cluster è attivo, anche se non viene effettivamente utilizzato. Per evitare di sostenere i costi del cluster, è necessario eliminarlo. Per informazioni, consulta [Fase 4: Eliminazione di un cluster](#).

Creazione di un cluster (API MemoryDB)

Per creare un cluster utilizzando l'API MemoryDB, usa l'azione. [CreateCluster](#)

Important

Non appena il cluster diventa disponibile, viene addebitata ogni ora o frazione di ora in cui il cluster è attivo, anche se non viene utilizzato. Per evitare di sostenere i costi del cluster, è necessario eliminarlo. Per informazioni, consulta [Fase 4: Eliminazione di un cluster](#).

Configurazione dell'autenticazione

Per informazioni sulla configurazione dell'autenticazione per il cluster, consulta [Autenticazione con IAM](#) e [Autenticazione degli utenti con elenchi di controllo degli accessi \(ACL\)](#).

Passaggio 2: autorizzare l'accesso al cluster

Questa sezione presuppone che l'utente abbia familiarità con l'avvio e la connessione a istanze Amazon EC2. Per ulteriori informazioni, consulta le [Guida di nozioni di base per Amazon EC2](#).

I cluster MemoryDB sono progettati per essere accessibili da un'istanza Amazon EC2. È inoltre possibile accedervi tramite applicazioni containerizzate o serverless in esecuzione in Amazon Elastic Container Service o AWS Lambda. Lo scenario più comune consiste nell'accedere a un cluster MemoryDB da un'istanza Amazon EC2 nello stesso Amazon Virtual Private Cloud (Amazon VPC), come nel caso di questo esercizio.

Prima di poter eseguire la connessione a un cluster da un'istanza EC2, devi autorizzare l'istanza EC2 ad accedere al cluster.

Il caso d'uso più comune è quando un'applicazione distribuita su un'istanza EC2 deve connettersi a un cluster nello stesso VPC. Il modo più semplice per gestire l'accesso tra istanze EC2 e istanze database nello stesso VPC consiste nel fare quanto segue:

1. Creare un gruppo di sicurezza VPC per il cluster. Questo gruppo di sicurezza può essere utilizzato per limitare l'accesso ai cluster. Per questo gruppo di sicurezza è ad esempio possibile creare una regola personalizzata che consenta l'accesso TCP tramite la porta assegnata al cluster al momento della creazione e un indirizzo IP che verrà utilizzato per accedere al cluster.

La porta predefinita per i cluster MemoryDB è 6379.

2. Creare un gruppo di sicurezza VPC per le istanze EC2 (server Web e di applicazioni). Questo gruppo di sicurezza può, se necessario, consentire l'accesso all'istanza EC2 da Internet tramite la tabella di routing del VPC. Ad esempio, può impostare regole in questo gruppo di sicurezza per consentire l'accesso TCP all'istanza EC2 sulla porta 22.
3. Crea regole personalizzate nel gruppo di sicurezza per il tuo cluster che consentano le connessioni dal gruppo di sicurezza che hai creato per le tue istanze EC2. Ciò consente a qualsiasi membro del gruppo di sicurezza di accedere ai cluster.

Per creare in un gruppo di sicurezza VPC una regola che consenta connessioni da un altro gruppo di sicurezza

1. [Accedi alla console di AWS gestione e apri la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc](https://console.aws.amazon.com/vpc).
2. Nel riquadro di navigazione a sinistra, scegli Security Groups (Gruppi di sicurezza).

3. Seleziona o crea un gruppo di sicurezza da utilizzare per i tuoi cluster. In Regole in entrata, scegliere Modifica regole in entrata e quindi Aggiungi regola. Tale gruppo di sicurezza consentirà di accedere ai membri di un altro gruppo di sicurezza.
4. In Type (Tipo) scegliere Custom TCP Rule (Regola TCP personalizzata).
 - a. Per Port Range (Intervallo porte) specificare la porta utilizzata alla creazione del cluster.

La porta predefinita per i cluster MemoryDB è. 6379
 - b. Nella casella Source (fonte) iniziare a digitare l'ID del gruppo di sicurezza. Nell'elenco selezionare il gruppo di sicurezza da utilizzare per le istanze Amazon EC2.
5. Scegliere Save (Salva) al termine.

Dopo aver abilitato l'accesso, è ora possibile connettersi al cluster, come illustrato nella sezione successiva.

Per informazioni sull'accesso al cluster MemoryDB da un altro Amazon VPC, da una AWS regione diversa o persino dalla rete aziendale, consulta quanto segue:

- [Modelli di accesso a un cluster MemoryDB in una Amazon VPC](#)
- [Accesso alle risorse MemoryDB dall'esternoAWS](#)

Fase 3: Connect al cluster

Prima di continuare, completa [Passaggio 2: autorizzare l'accesso al cluster](#).

Questa sezione presuppone che sia stata creata un'istanza Amazon EC2 ed eseguita la connessione alla stessa. Per istruzioni su come procedere, consulta la [Guida alle operazioni di base di Amazon EC2](#).

Un'istanza Amazon EC2 può connettersi a un cluster solo se l'hai autorizzata a farlo.

Trova il tuo endpoint del cluster

Quando lo stato del cluster è disponibile e l'accesso al cluster è stato autorizzato, puoi accedere a un'istanza Amazon EC2 e connetterti al cluster. A questo scopo, devi innanzitutto determinare l'endpoint.

Per ulteriori informazioni su come trovare gli endpoint, consulta quanto segue:

- [Ricerca dell'endpoint per un cluster MemoryDB \(AWS Management Console\)](#)
- [Ricerca dell'endpoint per un cluster MemoryDB \(AWSCLI\)](#)
- [Ricerca dell'endpoint per un cluster MemoryDB \(API MemoryDB\)](#)

Connect a un cluster MemoryDB (Linux)

Ora che hai l'endpoint di cui hai bisogno, puoi accedere a un'istanza EC2 e connetterti al cluster. Nell'esempio seguente, usi l'utilità cli per connetterti a un cluster usando Ubuntu 22. L'ultima versione di cli supporta anche SSL/TLS per la connessione di cluster abilitati alla crittografia/autenticazione.

Connessione ai nodi MemoryDB utilizzando redis-cli

Per accedere ai dati dai nodi MemoryDB, si utilizzano client che funzionano con Secure Socket Layer (SSL). Puoi anche usare redis-cli con TLS/SSL su Amazon linux e Amazon Linux 2.

Per utilizzare redis-cli per connettersi a un cluster MemoryDB su Amazon Linux 2 o Amazon Linux

1. Scaricare e compilare l'utilità redis-cli. Questa utilità è inclusa nella distribuzione del software Redis.
2. Al prompt dei comandi dell'istanza EC2, digita i comandi appropriati per la versione di Linux che stai utilizzando.

Amazon Linux 2023

Se usi Amazon Linux 2023, inserisci questo:

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

Quindi digita il comando seguente, sostituendo l'endpoint del cluster e la porta con quanto mostrato in questo esempio.

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

Per ulteriori informazioni sulla ricerca dell'endpoint, consulta [Individuazione degli endpoint dei nodi](#).

Amazon Linux 2

Se usi Amazon Linux 2, inserisci questo:

```
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

Se usi Amazon Linux, inserisci questo:

```
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/
```

Su Amazon Linux, potrebbe essere necessario eseguire anche i seguenti passaggi aggiuntivi:

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

3. Dopo aver scaricato e installato l'utilità redis-cli, si consiglia di eseguire il comando opzionale.
make-test
4. Per connetterti a un cluster con crittografia e autenticazione abilitate, inserisci questo comando:

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

Note

Se installi redis6 su Amazon Linux 2023, ora puoi usare redis6-cli il comando al posto di: redis-cli

```
redis6-cli -h Primary or Configuration Endpoint --tls -p 6379
```

Fase 4: Eliminazione di un cluster

Fintantoché un cluster è nello stato disponibile, ne vengono addebitati i costi, anche se non è utilizzato attivamente. Per interrompere l'addebito, elimina il cluster.

Warning

Quando si elimina un cluster MemoryDB, le istantanee manuali vengono conservate. È inoltre possibile creare uno snapshot finale prima che il cluster venga eliminato. Le istantanee automatiche non vengono conservate. Per ulteriori informazioni, consulta [Snapshot e ripristino](#).

Usando il AWS Management Console

La procedura seguente elimina un solo cluster dalla distribuzione. Per eliminare più cluster, ripetere la procedura per ogni cluster da eliminare. Non occorre attendere la fine dell'eliminazione di un cluster prima di avviare la procedura per eliminarne un altro.

Per eliminare un cluster

1. [Accedere AWS Management Console e aprire la console MemoryDB for Redis all'indirizzo `https://console.aws.amazon.com/memorydb/`.](https://console.aws.amazon.com/memorydb/)
2. Per scegliere il cluster da eliminare, scegli il pulsante di opzione accanto al nome del cluster dall'elenco dei cluster. In questo caso, sostituire con il nome del cluster creato in [Fase 1: creazione di un cluster](#).
3. In Actions (Azioni), scegliere Delete (Elimina).
4. Per prima cosa scegli se creare un'istantanea del cluster prima di eliminarlo, quindi inserisci `delete` nella casella di conferma ed elimina per eliminare il cluster, oppure scegli Annulla per conservare il cluster.

Se si sceglie Delete (Elimina), lo stato del cluster diventa in fase di eliminazione.

Non appena il cluster viene rimosso dall'elenco di cluster, non ti verranno più addebitati costi.

Usando il AWS CLI

Il seguente codice elimina il cluster `my-cluster`. In questo caso, sostituire `my-cluster` con il nome del cluster creato in [Fase 1: creazione di un cluster](#).

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

L'operazione `delete-cluster` CLI elimina solo un cluster. Per eliminare più cluster, chiama `delete-cluster` ogni cluster che desideri eliminare. Non è necessario attendere il completamento dell'eliminazione di un cluster prima di eliminarne un altro.

Per Linux, macOS o Unix:

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

Per Windows:

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

Per ulteriori informazioni, consulta [delete-cluster](#).

Utilizzo dell'API MemoryDB

Il seguente codice elimina il cluster `my-cluster`. In questo caso, sostituisci `my-cluster` con il nome del cluster creato in [Fase 1: creazione di un cluster](#).

```
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>
```

L'operazione `DeleteCluster` API elimina solo un cluster. Per eliminare più cluster, richiama `DeleteCluster` ogni cluster che desideri eliminare. Non è necessario attendere il completamento dell'eliminazione di un cluster prima di eliminarne un altro.

Per ulteriori informazioni, vedere [DeleteCluster](#)

A questo punto come si può procedere?

Dopo aver provato l'esercizio Guida introduttiva, puoi esplorare le seguenti sezioni per saperne di più su MemoryDB e sugli strumenti disponibili:

- [Guida introduttiva con AWS](#)
- [Strumenti per Amazon Web Services](#)
- [Interfaccia a riga di comando AWS](#)
- [Riferimento all'API MemoryDB per Redis.](#)

Gestione di nodi

Un nodo è l'elemento costitutivo più piccolo di una distribuzione MemoryDB per Redis. Un nodo appartiene a uno shard che appartiene a un cluster. Ogni nodo esegue la versione del motore scelta al momento della creazione o dell'ultima modifica del cluster. Ogni nodo dispone del proprio nome Domain Name Service (DNS) e porta. Sono supportati diversi tipi di nodi MemoryDB, ciascuno con quantità variabili di memoria e potenza di calcolo associate.

Argomenti

- [Nodi e shard di MemoryDB](#)
- [Tipi di nodi supportati](#)
- [Nodi riservati MemoryDB](#)
- [Sostituzione dei nodi](#)

Alcune importanti operazioni che coinvolgono i nodi sono le seguenti:

- [Aggiunta/rimozione di nodi da un cluster](#)
- [Dimensionamento](#)
- [Individuazione degli endpoint di connessione](#)

Nodi e shard di MemoryDB

Uno shard è una disposizione gerarchica di nodi, ciascuno racchiuso in un cluster. Le partizioni supportano la replica. All'interno di una partizione, un nodo funziona come il nodo primario di lettura/scrittura. Tutti gli altri nodi in una partizione funzionano come repliche di sola lettura del nodo primario. MemoryDB supporta più shard all'interno di un cluster. Questo supporto consente il partizionamento dei dati in un cluster MemoryDB.

MemoryDB supporta la replica tramite shard. L'operazione API [DescribeClusters](#) elenca gli shard con i nodi membri, i nomi dei nodi, gli endpoint e anche altre informazioni.

Dopo aver creato un cluster MemoryDB, può essere modificato (ridimensionato o ridimensionato). Per ulteriori informazioni, consultare [Dimensionamento](#) e [Sostituzione dei nodi](#).

Quando si crea un nuovo cluster, è possibile eseguirne il popolamento con i dati del vecchio cluster in modo che non inizi vuoto. Questa operazione può essere utile se devi modificare il tipo di nodo, la

versione del motore o migrare da Amazon ElastiCache for Redis. Per ulteriori informazioni, consultare [Creazione di istantanee manuali](#) e [Ripristino da uno snapshot](#).

Tipi di nodi supportati

MemoryDB supporta i seguenti tipi di nodi.

Memoria ottimizzata

Tipo di istanza	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)	Multiplexing I/O avanzato (Redis 7.0.4+)	Versione minima del motore
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	Sì	6.2
db.r7g.4xlarge	7,5	15	Sì	6.2
db.r7g.8xlarge	15	N/D	Sì	6.2
db.r7g.12xlarge	22,5	N/D	Sì	6.2
db.r7g.16xlarge	30	N/D	Sì	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	Sì	6.2
db.r6g.4xlarge	5.0	10,0	Sì	6.2
db.r6g.8xlarge	12	N/D	Sì	6.2
db.r6g.12xlarge	20	N/D	Sì	6.2
db.r6g.16xlarge	25	N/D	Sì	6.2

Memoria ottimizzata con il tiering dei dati

Tipo di istanza	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)	Multiplexing I/O avanzato (Redis 7.0.4+)	Versione minima del motore
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/D	No	6.2

Nodi per uso generico

Tipo di istanza	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)	Multiplexing I/O avanzato (Redis 7.0.4+)	Versione minima del motore
db.t4g.small	0,128	5.0	No	6.2
db.t4g.medium	0,256	5.0	No	6.2

Per informazioni sulla disponibilità AWS regionale, consulta i prezzi di [MemoryDB](#) for Redis

Tutti i tipi di nodi vengono creati in un cloud privato virtuale (VPC).

Nodi riservati MemoryDB

I nodi riservati offrono uno sconto significativo rispetto ai prezzi dei nodi on demand. I nodi riservati non sono nodi fisici, ma piuttosto uno sconto di fatturazione applicato all'uso di nodi on-demand nel tuo account. Gli sconti per i nodi riservati sono legati al tipo di nodo e AWS alla regione.

La procedura generale per l'utilizzo dei nodi riservati è la seguente:

- Consulta le informazioni sulle offerte disponibili di nodi riservati
- Acquista un'offerta di nodi riservati utilizzando AWS Command Line Interface o AWS Management Console SDK
- Consulta le informazioni sui nodi riservati esistenti

Argomenti

- [Panoramica dei nodi riservati](#)

Panoramica dei nodi riservati

Quando acquisti un nodo riservato di MemoryDB, ti impegni a ottenere una tariffa scontata, per un tipo di nodo specifico, per la durata del nodo riservato. Per utilizzare un nodo riservato MemoryDB, si crea un nuovo nodo proprio come si fa per un nodo su richiesta. Il nuovo nodo creato deve corrispondere alle specifiche del nodo riservato. Se le specifiche del nuovo nodo corrispondono a un nodo riservato esistente per il tuo account, ti verrà addebitata la tariffa scontata offerta per il nodo riservato. Altrimenti, il nodo viene fatturato a una tariffa su richiesta. Puoi utilizzare l'API AWS Management Console AWS CLI, the o MemoryDB per elencare e acquistare le offerte di nodi riservati disponibili.

MemoryDB offre nodi riservati per i nodi R7g, R6g e R6gd ottimizzati per la memoria (con tiering dei dati). [Per informazioni sui prezzi, consulta la pagina dei prezzi di MemoryDB for Redis.](#)

Tipi offerta

I nodi riservati sono disponibili in tre varietà: No Upfront, Partial Upfront e All Upfront, che consentono di ottimizzare i costi di MemoryDB for Redis in base all'utilizzo previsto.

No Upfront: questa opzione fornisce l'accesso a un nodo riservato senza richiedere un pagamento anticipato. Il tuo nodo riservato No Upfront fattura una tariffa oraria scontata per ogni ora entro il termine, indipendentemente dall'utilizzo, e non è richiesto alcun pagamento anticipato.

Pagamento anticipato parziale: questa opzione richiede il pagamento anticipato di una parte del nodo riservato. Le ore rimanenti del periodo di prenotazione vengono fatturate a una tariffa oraria scontata, indipendentemente dall'utilizzo.

Tutto anticipato: il pagamento completo viene effettuato all'inizio del periodo, senza altri costi sostenuti per il resto del periodo, indipendentemente dal numero di ore utilizzate.

Tutti e tre i tipi di offerta sono disponibili per periodi di un anno e tre anni.

Dimensioni dei nodi riservati flessibili

Quando acquisti un nodo riservato, una cosa che specifichi è il tipo di nodo, ad esempio `db.r6g.xlarge`. [Per ulteriori informazioni sui tipi di nodi, consulta MemoryDB for Redis Pricing.](#)

Se disponi di un nodo e devi scalarlo per renderlo più capiente, il nodo riservato viene automaticamente applicato al nodo scalato. In altre parole, i nodi riservati vengono applicati automaticamente all'utilizzo di qualsiasi dimensione nella stessa famiglia di nodi. I nodi riservati con dimensioni flessibili sono disponibili per i nodi con la stessa AWS regione. I nodi riservati flessibili in termini di dimensioni possono scalare solo nelle rispettive famiglie di nodi. Ad esempio, un nodo riservato per un `db.r6g.xlarge` può essere applicato a un `db.r6g.2xlarge`, ma non a un `db.r6gd.large`, perché `db.r6g` e `db.r6gd` sono famiglie di nodi diverse.

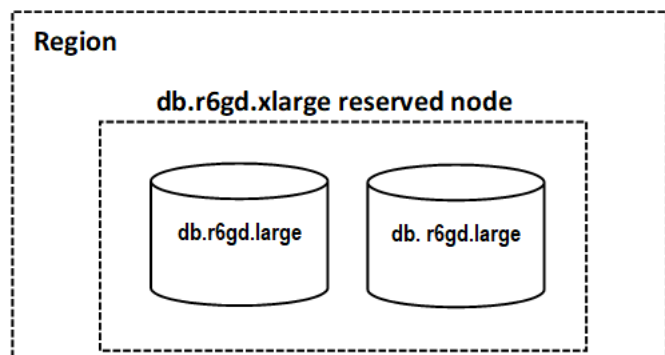
La flessibilità delle dimensioni significa che è possibile spostarsi liberamente tra le configurazioni all'interno della stessa famiglia di nodi. Ad esempio, è possibile passare da un nodo riservato `r6g.xlarge` (8 unità normalizzate) a due nodi riservati `r6g.large` (8 unità normalizzate) ($2 \times 4 = 8$ unità normalizzate) nella stessa regione senza costi aggiuntivi. AWS

È possibile confrontare l'utilizzo per diverse dimensioni dei nodi riservati utilizzando unità normalizzate. Ad esempio, un'unità di utilizzo su due nodi `db.r6g.4xlarge` equivale a 16 unità di utilizzo normalizzate su un `db.r6g.large`. La tabella seguente mostra il numero di unità normalizzate per ogni dimensione del nodo:

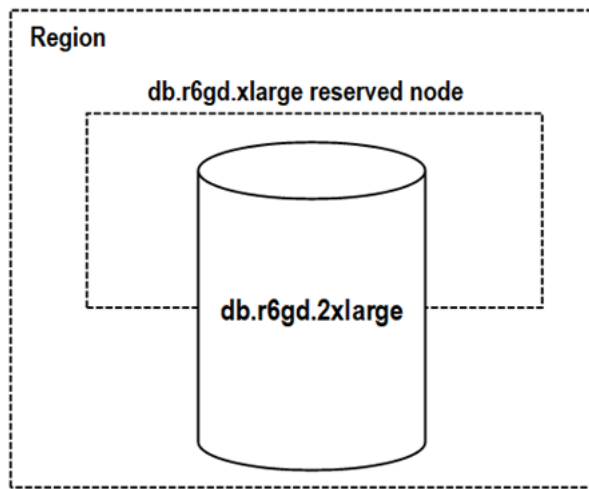
Dimensioni nodo	Unità normalizzate
small	1
medium	2
large	4

Dimensioni nodo	Unità normalizzate
xlarge	8
2xlarge	16
4xlarge	32
6xlarge	48
8xlarge	64
10xlarge	80
12xlarge	96
16xlarge	128

Ad esempio, acquisti un nodo riservato `db.r6gd.xlarge` e hai due nodi riservati `db.r6gd.large` in esecuzione nel tuo account nella stessa regione. AWS In questo caso, il vantaggio di fatturazione viene applicato integralmente a entrambi i nodi.



In alternativa, se hai un'istanza `db.r6gd.2xlarge` in esecuzione nel tuo account nella stessa AWS regione, il vantaggio di fatturazione viene applicato al 50 percento dell'utilizzo del nodo riservato.



Eliminazione di un nodo riservato

I termini per un nodo riservato prevedono un impegno di un anno o tre anni. Non puoi annullare un nodo riservato. Tuttavia, puoi eliminare un nodo coperto da uno sconto per i nodi riservati. Il processo di eliminazione di un nodo coperto da uno sconto sui nodi riservati è lo stesso di qualsiasi altro nodo.

Se elimini un nodo coperto da uno sconto sui nodi riservati, puoi avviare un altro nodo con specifiche compatibili. In questo caso, continuare a usufruire della tariffa scontata durante il periodo della prenotazione (un anno o tre anni).

Lavorare con nodi riservati

È possibile utilizzare l'API AWS Management Console AWS Command Line Interface, the e MemoryDB per lavorare con nodi riservati.

Console

Per ottenere prezzi e informazioni sulle offerte disponibili di nodi riservati

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Nel riquadro di navigazione, scegli Nodi riservati.
3. Scegli Acquista nodi riservati.
4. Per Tipo di nodo, scegli il tipo di nodo che desideri distribuire.
5. Per Quantità, scegli il numero di nodi che desideri distribuire.
6. In Term, scegli per quanto tempo desideri riservare il nodo del database.

7. Per Offering type (Tipo di offerta), scegliere il tipo di offerta.

Dopo aver effettuato queste selezioni, puoi visualizzare le informazioni sui prezzi nella sezione Riepilogo della prenotazione.

 Important

Scegli Annulla per evitare di acquistare questi nodi riservati e di incorrere in addebiti.

Dopo aver ottenuto informazioni sulle offerte disponibili per i nodi riservati, puoi utilizzare le informazioni per acquistare un'offerta come illustrato nella procedura seguente:

Per acquistare un nodo riservato

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Nel riquadro di navigazione, scegli Nodi riservati.
3. Scegli Acquista nodi riservati.
4. Per Tipo di nodo, scegli il tipo di nodo che desideri distribuire.
5. Per Quantità, scegli il numero di nodi che desideri distribuire.
6. In Term, scegli per quanto tempo desideri riservare il nodo del database.
7. Per Offering type (Tipo di offerta), scegliere il tipo di offerta.
8. (Facoltativo) Puoi assegnare il tuo identificatore ai nodi riservati acquistati per aiutarti a tracciarli. Per ID di prenotazione, digita un identificatore per il tuo nodo riservato.

Dopo aver effettuato queste selezioni, puoi visualizzare le informazioni sui prezzi nella sezione Riepilogo della prenotazione.

9. Scegli Acquista nodi riservati.
10. I nodi riservati vengono acquistati e quindi visualizzati nell'elenco dei nodi riservati.

Per ottenere informazioni sui nodi riservati per il tuo AWS account

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Nel riquadro di navigazione, scegli Nodi riservati.

3. Vengono visualizzati i nodi riservati per il tuo account. Per visualizzare informazioni dettagliate su un particolare nodo riservato, scegli quel nodo nell'elenco. È quindi possibile visualizzare informazioni dettagliate su quel nodo nei dettagli.

AWS Command Line Interface

L'output di `aws memorydb describe-reserved-nodes-offerings` seguente restituisce i dettagli delle offerte relative ai nodi riservati.

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

Questo produce un output simile al seguente:

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

Puoi anche passare i seguenti parametri per limitare l'ambito di ciò che viene restituito:

- `--reserved-nodes-offering-id` – L'ID dell'offerta da acquistare.
- `--node-type`— Il valore del filtro del tipo di nodo. Utilizzate questo parametro per mostrare solo le prenotazioni che corrispondono al tipo di nodo specificato.
- `--duration`— Il valore del filtro della durata, specificato in anni o secondi. Usa questo parametro per mostrare solo le prenotazioni per questa durata.

- `--offering-type`— Utilizzate questo parametro per mostrare solo le offerte disponibili che corrispondono al tipo di offerta specificato.

Dopo aver ottenuto informazioni sulle offerte disponibili dei nodi riservati, è possibile utilizzare le informazioni per acquistare un'offerta.

L'`purchase-reserved-nodes-offering` seguente acquista nuovi nodi riservati

Per 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
```

Per Windows:

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

- `--reserved-nodes-offering-id` rappresenta il nome dei nodi riservati che offrono l'acquisto.
- `--reservation-id` è un identificativo specificato dal cliente per tracciare questa prenotazione.

Note

L'ID di prenotazione è un identificativo univoco specificato dal cliente per tracciare questa prenotazione. Se questo parametro non è specificato, MemoryDB genera automaticamente un identificatore per la prenotazione.

- `--node-count` è il numero di nodi da prenotare. Il valore predefinito è 1.

Questo produce un output simile al seguente:

```
{  
  "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:xxxxxxx:reservednode/reservation"
  }
}

```

Dopo aver acquistato i nodi riservati, è possibile ottenere informazioni sui nodi riservati.

L'operazione `describe-reserved-nodes` seguente restituisce informazioni sui nodi riservati per questo account.

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

Questo produce un output simile al seguente:

```

{
  "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:xxxxxxx:reservednode/
ri-2022-12-16-00-28-40-600"
    }
]
}

```

Puoi anche passare i seguenti parametri per limitare l'ambito di ciò che viene restituito:

- `--reservation-id`— È possibile assegnare il proprio identificatore ai nodi riservati acquistati per facilitarne il tracciamento.
- `--reserved-nodes-offering-id`— Il valore del filtro dell'identificatore dell'offerta. Utilizza questo parametro per mostrare solo le prenotazioni acquistate che corrispondono all'identificativo dell'offerta specificato.
- `--node-type`— Il valore del filtro del tipo di nodo. Utilizzate questo parametro per mostrare solo le prenotazioni che corrispondono al tipo di nodo specificato.
- `--duration`— Il valore del filtro della durata, specificato in anni o secondi. Usa questo parametro per mostrare solo le prenotazioni per questa durata.
- `--offering-type`— Utilizzate questo parametro per mostrare solo le offerte disponibili che corrispondono al tipo di offerta specificato.

API MemoryDB

I seguenti esempi mostrano come utilizzare l'[API MemoryDB Query](#) per i nodi riservati:

DescribeReservedNodesOfferings

Restituisce i dettagli delle offerte relative ai nodi riservati.

```

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>

```

I seguenti parametri limitano l'ambito di ciò che viene restituito:

- **ReservedNodesOfferingId** rappresenta il nome dei nodi riservati che offrono l'acquisto.
- **Duration**— Il valore del filtro di durata, specificato in anni o secondi. Usa questo parametro per mostrare solo le prenotazioni per questa durata.
- **NodeType**— Il valore del filtro del tipo di nodo. Utilizzate questo parametro per mostrare solo le offerte che corrispondono al tipo di nodo specificato.
- **OfferingType**— Utilizzate questo parametro per mostrare solo le offerte disponibili che corrispondono al tipo di offerta specificato.

Dopo aver ottenuto informazioni sulle offerte disponibili dei nodi riservati, è possibile utilizzare le informazioni per acquistare un'offerta.

PurchaseReservedNodesOffering

Consente di acquistare un'offerta di nodi riservati.

```

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` rappresenta il nome dei nodi riservati che vengono offerti in acquisto.
- `ReservationID` è un identificativo specificato dal cliente per tracciare questa prenotazione.

Note

L'ID di prenotazione è un identificativo univoco specificato dal cliente per tracciare questa prenotazione. Se questo parametro non è specificato, MemoryDB genera automaticamente un identificatore per la prenotazione.

- `NodeCount` è il numero di nodi da riservare. Il valore predefinito è 1.

Dopo aver acquistato i nodi riservati, è possibile ottenere informazioni sui nodi riservati.

DescribeReservedNodes

Restituisce informazioni sui nodi riservati per questo account.

```
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>
```

I seguenti parametri limitano l'ambito di ciò che viene restituito:

- `ReservedNodesOfferingId` rappresenta il nome del nodo riservato.
- `ReservationID`— È possibile assegnare il proprio identificatore ai nodi riservati acquistati per facilitarne il tracciamento.

- **NodeType**— Il valore del filtro del tipo di nodo. Utilizzate questo parametro per mostrare solo le prenotazioni che corrispondono al tipo di nodo specificato.
- **Duration**— Il valore del filtro della durata, specificato in anni o secondi. Usa questo parametro per mostrare solo le prenotazioni per questa durata.
- **OfferingType**— Utilizzate questo parametro per mostrare solo le offerte disponibili che corrispondono al tipo di offerta specificato.

Visualizzazione della fatturazione per i nodi riservati

Puoi visualizzare la fatturazione per i tuoi nodi riservati nella Dashboard di fatturazione in AWS Management Console

Per visualizzare la fatturazione dei nodi riservati

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Dal pulsante Cerca nella parte superiore della console, scegli Fatturazione.
3. Scegli Fatture dal lato sinistro della dashboard.
4. In Costi AWS di servizio, espandi MemoryDB.
5. Espandi la AWS regione in cui si trovano i tuoi nodi riservati, ad esempio Stati Uniti orientali (Virginia settentrionale).

I tuoi nodi riservati e le relative tariffe orarie per il mese corrente sono indicati nella sezione Istanze riservate di Amazon MemoryDB CreateCluster .

Amazon MemoryDB CreateCluster Reserved Instances		Hourly Fee
AmazonMemoryDB, db.r6g.large reserved instance applied	81.000 Hrs	\$0.00
AmazonMemoryDB, db.r6g.4xlarge reserved instance applied	324.000 Hrs	\$0.00
AmazonMemoryDB, db.r6g.4xlarge reserved instance applied	162.000 Hrs	\$0.00
USD hourly fee per AmazonMemoryDB, db.r6g.large instance	1,488.000 Hrs	\$0.00
USD hourly fee per AmazonMemoryDB, db.r6gd.2xlarge instance	744.000 Hrs	\$0.00
USD hourly fee per AmazonMemoryDB, db.r6g.4xlarge instance	744.000 Hrs	\$0.00
USD hourly fee per AmazonMemoryDB, db.r6gd.xlarge instance	744.000 Hrs	\$0.00
USD hourly fee per AmazonMemoryDB, db.r6gd.4xlarge instance	2,976.000 Hrs	\$0.00

Sostituzione dei nodi

MemoryDB aggiorna frequentemente la propria flotta con patch e aggiornamenti, di solito senza problemi. Tuttavia, di tanto in tanto dobbiamo riavviare i nodi MemoryDB per applicare gli aggiornamenti obbligatori del sistema operativo all'host sottostante. Queste sostituzioni sono necessarie per l'applicazione di aggiornamenti intesi a rafforzare sicurezza, affidabilità e prestazioni.

Offriamo l'opzione per gestire personalmente la sostituzione dei nodi nel momento che si ritiene più adatto, prima della finestra di sostituzione nodo pianificata. Quando la gestione della sostituzione è manuale, l'istanza riceve l'aggiornamento del sistema operativo quando il nodo viene riavviato e la finestra di sostituzione del nodo programmata viene annullata. Potresti continuare a ricevere avvisi che indicano l'esecuzione dell'attività di sostituzione del nodo. Se hai già ridotto manualmente la necessità di manutenzione, puoi ignorare questi avvisi.

Note

I nodi sostitutivi generati automaticamente da MemoryDB for Redis possono avere indirizzi IP diversi. L'utente è responsabile della revisione della configurazione dell'applicazione per assicurarsi che i nodi siano associati agli indirizzi IP appropriati.

L'elenco seguente identifica le azioni che puoi intraprendere quando MemoryDB pianifica la sostituzione di uno dei tuoi nodi:

Opzioni di sostituzione dei nodi MemoryDB

- Non fare nulla: se non si fa nulla, MemoryDB sostituisce il nodo come pianificato.

Se il nodo è membro di un cluster Multi-AZ, MemoryDB offre una maggiore disponibilità durante l'applicazione di patch, aggiornamenti e altre sostituzioni dei nodi relative alla manutenzione.

La sostituzione viene completata mentre il cluster soddisfa le richieste di scrittura in entrata.

- Modifica della finestra di manutenzione: per gli eventi di manutenzione pianificata, si riceve un'e-mail o un evento di notifica da MemoryDB. In questi casi, se si modifica la finestra di manutenzione prima dell'orario di sostituzione pianificato, il nodo viene sostituito al nuovo orario. Per ulteriori informazioni, consulta [Modifica di un cluster MemoryDB](#).

Note

La possibilità di modificare la finestra di sostituzione spostando la finestra di manutenzione è disponibile solo quando la notifica di MemoryDB include una finestra di manutenzione. Se la notifica non include una finestra di manutenzione, non puoi modificare la finestra di sostituzione.

Supponiamo, ad esempio, che sia giovedì 9 novembre alle 15:00 e che la prossima finestra di manutenzione sia prevista per venerdì 10 novembre alle 17:00. Questi sono tre possibili scenari e i relativi risultati:

- Sposti la finestra di manutenzione alle ore 16:00 di venerdì, dopo la data e l'ora corrente e prima della prossima finestra di manutenzione programmata. Il nodo viene sostituito venerdì 10 novembre alle 16:00.
- Sposti la finestra di manutenzione alle ore 16:00 di sabato, dopo la data e l'ora corrente e dopo la prossima finestra di manutenzione programmata. Il nodo viene sostituito sabato 11 novembre alle 16:00.
- La finestra di manutenzione viene impostata su mercoledì alle 16:00, prima della settimana rispetto alla data e all'ora correnti. Il nodo viene sostituito il prossimo mercoledì 15 novembre alle 16:00.

Per istruzioni, consultare [Gestione della manutenzione](#).

Gestione dei cluster

La maggior parte delle operazioni di MemoryDB viene eseguita a livello di cluster. Un cluster può essere configurato con un numero specifico di nodi e con un gruppo di parametri che controlla le proprietà di ciascun nodo. Tutti i nodi all'interno di un cluster sono progettati in modo da essere dello stesso tipo e da avere le stesse impostazioni del gruppo di parametri e del gruppo di sicurezza.

Ogni cluster deve avere un proprio identificatore. L'identificatore del cluster è un nome fornito dal cliente. Questo identificatore specifica un particolare cluster quando interagisce con l'API e i comandi di MemoryDB. AWS CLI L'identificatore del cluster deve essere univoco per il cliente in una regione AWS.

I cluster MemoryDB sono progettati per essere accessibili utilizzando un'istanza Amazon EC2. Puoi avviare il tuo cluster MemoryDB solo in un cloud privato virtuale (VPC) basato sul servizio Amazon VPC, ma puoi accedervi dall'esterno. AWS Per ulteriori informazioni, consulta [Accesso alle risorse MemoryDB dall'esternoAWS](#).

Tiering di dati

I cluster che utilizzano un tipo di nodo dalla famiglia r6gd hanno i loro dati tiering di dati tra memoria e archiviazione SSD (Solid State Drive) locale. Il tiering di dati fornisce una nuova opzione prezzo-prestazioni per i carichi di lavoro Redis utilizzando Solid State Drive (SSD) a basso costo in ogni nodo di cluster oltre all'archiviazione dei dati in memoria. Analogamente ad altri tipi di nodi, i dati scritti sui nodi r6gd vengono archiviati in modo duraturo in un registro delle transazioni Multi-AZ. Il tiering dei dati è ideale per carichi di lavoro che accedono regolarmente fino al 20% del set di dati complessivo e per applicazioni che possono tollerare una latenza supplementare durante l'accesso ai dati su SSD.

Sui cluster con tiering di dati, MemoryDB monitora l'ora dell'ultimo accesso di ogni elemento archiviato. Quando la memoria disponibile (DRAM) è completamente esaurita, quando si accede successivamente ai dati su SSD, MemoryDB li sposta in modo automatico e asincrono nella memoria prima di elaborare la richiesta. Se si dispone di un carico di lavoro che accede regolarmente a un sottoinsieme di dati, il tiering di dati è un modo ottimale per dimensionare la capacità a costi contenuti.

Tieni presente che quando utilizzi il tiering dei dati, le chiavi rimangono sempre in memoria, mentre il posizionamento dei valori sulla memoria viene gestito da LRU e non dal disco. In generale, è preferibile che le dimensioni delle chiavi siano inferiori a quelle dei valori quando utilizzi il tiering dei dati.

Il tiering dei dati è progettato per avere un impatto minimo sulle prestazioni dei carichi di lavoro delle applicazioni. Ad esempio, supponendo valori String a 500 byte, in genere è possibile prevedere ulteriori 450 microsecondi di latenza per le richieste di lettura su SSD rispetto alle richieste di lettura ai dati in memoria.

Con le dimensioni più grandi dei nodi di tiering di dati (db.r6gd.8xlarge), è possibile archiviare fino a ~ 500 TB in un singolo cluster a 500 nodi (250 TB quando si utilizza 1 replica di lettura). Per il tiering dei dati, MemoryDB riserva il 19% della memoria (DRAM) per nodo per uso non legato ai dati. Il tiering di dati è compatibile con tutti i comandi e le strutture dati di Redis supportati in MemoryDB. Non è necessaria alcuna modifica lato client per utilizzare questa caratteristica.

Argomenti

- [Best practice](#)
- [Restrizioni](#)
- [Prezzi del tiering di dati](#)

- [Monitoraggio](#)
- [Utilizzo del tiering di dati](#)
- [Ripristino di dati da da da da un'istantanea su cluster con tiering di dati abil\(tiering di dati abil\(tiering di dati abil](#)

Best practice

È preferibile seguire le best practice seguenti:

- Il tiering dei dati è ideale per carichi di lavoro che accedono regolarmente fino al 20% del set di dati complessivo e per applicazioni che possono tollerare una latenza supplementare durante l'accesso ai dati su SSD.
- Quando utilizzi la capacità SSD disponibile su nodi con tiering di dati, è ti consigliamo una dimensione del valore superiore a quella della chiave. La dimensione del valore non può essere superiore a 128 MB; altrimenti non verrà spostato su disco. Quando gli elementi vengono spostati tra DRAM e SSD, le chiavi rimarranno sempre in memoria e solo i valori verranno spostati al livello SSD.

Restrizioni

Il livello di dati presenta le seguenti limitazioni:

- Il tipo di nodo utilizzato deve appartenere alla famiglia r6gd, disponibile nelle regioni seguenti: 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 e sa-east-1.
- Non è possibile ripristinare un'istantanea di un cluster r6gd in un altro cluster a meno che non utilizzi anche r6gd.
- Non è possibile esportare un'istantanea su Amazon S3 per cluster con tiering di dati.
- Il salvataggio senza fork non è supportato.
- Il dimensionamento non è supportato da dati un cluster di tiering di dati (ad esempio, un cluster che utilizza un tipo di nodo r6gd) a un cluster che non utilizza il tiering di dati (ad esempio, un cluster che utilizza un tipo di nodo r6g).
- Il tiering di dati supporta solo policy maxmemory volatile-lru, allkeys-lru e noeviction.

- Gli elementi più grandi di 128 MiB non vengono spostati su SSD.

Prezzi del tiering di dati

I nodi R6gd hanno una capacità totale 5 volte superiore (memoria + SSD) e possono contribuire a un risparmio di oltre il 60% sui costi di archiviazione quando si esegue al massimo utilizzo rispetto ai nodi R6g (solo memoria) Per ulteriori informazioni, consulta la pagina dei [prezzi di MemoryDB](#).

Monitoraggio

MemoryDB offre metriche specificamente progettate per monitorare i cluster di prestazioni che utilizzano il tiering di dati. Per monitorare il rapporto tra elementi in DRAM rispetto a SSD, puoi utilizzare la `CurrItems` metrica in [Metriche per MemoryDB](#). Puoi calcolare la percentuale come: $(\text{CurrItems with Dimension: Tier = Memory} * 100) / (\text{CurrItems with no dimension filter})$. Quando la percentuale di elementi in memoria scende al di sotto del 5%, è preferibile valutare [Dimensionamento di cluster di MemoryDB](#).

Per ulteriori informazioni, consulta Parametri per i cluster MemoryDB che utilizzano il tiering di dati su [Metriche per MemoryDB](#).

Utilizzo del tiering di dati

Utilizzo del tiering di dati mediante AWS Management Console

Quando si crea un cluster, il tiering di dati viene utilizzato selezionando un tipo di nodo dalla famiglia r6gd, ad esempio db.r6gd.xlarge. La selezione di quel tipo di nodo abilita automaticamente il tiering di dati.

Per ulteriori informazioni sulla creazione di cluster, consulta [Fase 1: creazione di un cluster](#).

Abilitazione di più tiering di dati mediante AWS CLI

Quando si crea un cluster mediante AWS CLI, il tiering di dati viene utilizzato selezionando un tipo di nodo dalla famiglia r6gd, ad esempio db.r6gd.xlarge e impostando il `--data-tiering` parametro.

Non è possibile disattivare il tiering di dati quando si seleziona un tipo di nodo dalla famiglia r6gd. Se si imposta il parametro `--no-data-tiering`, l'operazione avrà esito negativo.

Per 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
```

Per Windows:

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

Dopo aver eseguito questa operazione, verrà visualizzata una risposta simile alla seguente:

```
{  
  "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:xxxxxxxxxxxx:cluster/my-cluster",  
    "SnapshotRetentionLimit": 0,  
    "MaintenanceWindow": "wed:03:00-wed:04:00",  
    "SnapshotWindow": "04:30-05:30",  
    "ACLName": "my-acl",  
    "DataTiering": "true",  
    "AutoMinorVersionUpgrade": true  
  }  
}
```

}

Ripristino di dati da da da da un'istantanea su cluster con tiering di dati abil(tiering di dati abil(tiering di dati abil

È possibile ripristinare un'istantanea su un nuovo cluster con tiering di dati abil(Console), (AWSCLI) o (API MemoryDB). Quando si crea un cluster utilizzando tipi di nodo nella famiglia r6gd, il tiering di dati è abilitato.

Ripristino di dati da da da un'istantanea su cluster con tiering di dati abil(console)

Per ripristinare un'istantanea su un nuovo cluster con tiering di dati abil(console)[Ripristino da un'istantanea \(console\)](#)

Nota che per abilitare il tiering di dati, è necessario selezionare un tipo di nodo dalla famiglia r6gd.

Ripristino di dati da da da un'istantanea su cluster con tiering di dati abil(AWS CLI)

Quando si crea un cluster medianteAWS CLI, il tiering di dati viene utilizzato per impostazione predefinita selezionando un tipo di nodo dalla famiglia r6gd, ad esempio db.r6gd.xlarge e impostando il `--data-tiering` parametro.

Non è possibile disattivare il tiering di dati quando si seleziona un tipo di nodo dalla famiglia r6gd. Se si imposta il parametro `--no-data-tiering`, l'operazione avrà esito negativo.

Per 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
```

Per 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
```

Dopo aver eseguito questa operazione, verrà visualizzata una risposta simile alla seguente:

```
{
  "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:xxxxxxxxxxxx:cluster/my-cluster",
    "SnapshotRetentionLimit": 0,
    "MaintenanceWindow": "wed:03:00-wed:04:00",
    "SnapshotWindow": "04:30-05:30",
    "ACLName": "my-acl",
    "DataTiering": "true"
  }
}
```

Preparazione di un cluster

Di seguito, puoi trovare le istruzioni per creare un cluster utilizzando la console MemoryDB, o l'API MemoryDB. AWS CLI

Ogni volta che si crea un cluster, è consigliabile eseguire alcuni lavori preparatori in modo da non dover aggiornare o apportare modifiche immediatamente.

Argomenti

- [Determina i tuoi requisiti](#)

Determina i tuoi requisiti

Preparazione

Conoscere le risposte alle seguenti domande aiuta a velocizzare la creazione del cluster:

- Assicurati di creare un gruppo di sottoreti nello stesso VPC prima di iniziare a creare un cluster. In alternativa, è possibile utilizzare il gruppo di sottoreti predefinito fornito. Per ulteriori informazioni, consulta [Sottoreti e gruppi di sottoreti](#).

MemoryDB è progettato per essere accessibile dall'interno AWS tramite Amazon EC2. Tuttavia, se si avvia in un VPC basato su Amazon VPC, è possibile fornire l'accesso dall'esterno. AWS Per ulteriori informazioni, consulta [Accesso alle risorse MemoryDB dall'esternoAWS](#).

- Devi personalizzare qualche valore di parametro?

In tal caso, crea un gruppo di parametri personalizzato. Per ulteriori informazioni, consulta [Creazione di un gruppo di parametri](#).

- Devi creare un gruppo di sicurezza VPC?

Per ulteriori informazioni, consulta [Security in Your VPC](#).

- Come intendi implementare la tolleranza ai guasti?

Per ulteriori informazioni, consulta [Limitazione dell'impatto degli errori](#).

Argomenti

- [Requisiti di memoria e del processore](#)
- [Configurazione del cluster MemoryDB](#)
- [Multiplexing I/O avanzato](#)
- [Requisiti di dimensionamento](#)
- [Requisiti di accesso](#)
- [Regione e zone di disponibilità](#)

Requisiti di memoria e del processore

L'elemento costitutivo di base di MemoryDB per Redis è il nodo. I nodi sono configurati in frammenti per formare cluster. Quando determini il tipo di nodo da utilizzare per il cluster, prendi in considerazione la configurazione dei nodi del cluster e la quantità di dati da archiviare.

Configurazione del cluster MemoryDB

I cluster MemoryDB sono composti da 1 a 500 shard. I dati in un cluster MemoryDB sono partizionati tra gli shard del cluster. L'applicazione si connette a un cluster MemoryDB utilizzando un indirizzo di rete chiamato Endpoint. Oltre agli endpoint del nodo, lo stesso cluster MemoryDB dispone di un endpoint chiamato endpoint del cluster. L'applicazione può utilizzare questo endpoint per leggere o scrivere nel cluster, lasciando a MemoryDB la determinazione del nodo da cui leggere o scrivere.

Multiplexing I/O avanzato

Se utilizzi la versione 7.0 o successiva di Redis, otterrai un'ulteriore accelerazione grazie al multiplexing I/O avanzato, in cui ogni thread IO di rete dedicato trasferisce i comandi da più client al motore Redis, sfruttando la capacità di Redis di elaborare in modo efficiente i comandi in batch. [Per ulteriori informazioni, consulta Prestazioni ultraveloci e. the section called “Tipi di nodi supportati”](#)

Requisiti di dimensionamento

Tutti i cluster possono essere scalati verso un tipo di nodo più grande. Quando si esegue il ridimensionamento di un cluster MemoryDB, è possibile farlo online in modo che il cluster rimanga disponibile oppure è possibile eseguire il seeding di un nuovo cluster da un'istantanea ed evitare che il nuovo cluster sia inizialmente vuoto.

Per ulteriori informazioni sul tagging, consulta [Dimensionamento](#) in questa guida.

Requisiti di accesso

In base alla progettazione, è possibile accedere ai cluster MemoryDB dalle istanze Amazon EC2. L'accesso di rete a un cluster MemoryDB è limitato all'account che ha creato il cluster. Pertanto, prima di poter accedere a un cluster da un'istanza Amazon EC2, è necessario autorizzare l'accesso al cluster. Per istruzioni dettagliate, consultare [Passaggio 2: autorizzare l'accesso al cluster](#) in questa guida.

Regione e zone di disponibilità

Posizionando i cluster di MemoryDB in una AWS regione vicina all'applicazione, è possibile ridurre la latenza. Se il cluster dispone di più nodi, posizionarli in zone di disponibilità diverse può ridurre l'effetto degli errori sul cluster.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Scelta di regioni e zone di disponibilità](#)
- [Limitazione dell'impatto degli errori](#)

Creazione di un cluster

MemoryDB for Redis offre tre modi per creare un cluster. Per ulteriori informazioni, consulta [Fase 1: creazione di un cluster](#).

Visualizzazione dei dettagli di un cluster

È possibile visualizzare informazioni dettagliate su uno o più cluster utilizzando la console MemoryDB o l'API MemoryDB. AWS CLI

Visualizzazione dei dettagli per un cluster MemoryDB (Console)

La procedura seguente descrive come visualizzare i dettagli di un cluster MemoryDB utilizzando la console MemoryDB.

1. [Accedere AWS Management Console e aprire la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. Per visualizzare i dettagli di un cluster, scegli il pulsante di opzione a sinistra del nome del cluster, quindi scegli Visualizza dettagli. Puoi anche fare clic direttamente sul cluster per visualizzare la pagina dei dettagli del cluster.

La pagina dei dettagli del cluster mostra i dettagli sul cluster, incluso l'endpoint del cluster. È possibile visualizzare ulteriori dettagli utilizzando le diverse schede disponibili nella pagina dei dettagli del cluster.

3. Scegli la scheda Frammenti e nodi per visualizzare un elenco degli shard del cluster e il numero di nodi in ogni shard.
4. Per visualizzare informazioni specifiche su un nodo, espandi lo shard nella tabella seguente. In alternativa, puoi anche cercare lo shard utilizzando la casella di ricerca.

In questo modo vengono visualizzate informazioni su ciascun nodo, tra cui la zona di disponibilità, gli slot/spazi chiave e lo stato.

5. Scegli la scheda Metriche per monitorare i rispettivi processi, come l'utilizzo della CPU e l'utilizzo della CPU del motore. Per ulteriori informazioni, consulta [Metriche per MemoryDB](#).
6. Scegli la scheda Rete e sicurezza per visualizzare i dettagli del gruppo di sottoreti e dei gruppi di sicurezza.
 - a. Nel gruppo di sottoreti, puoi visualizzare il nome del gruppo di sottoreti, un collegamento al VPC a cui appartiene la sottorete e l'Amazon Resource Name (ARN) del gruppo di sottoreti.
 - b. Nei gruppi di sicurezza, puoi visualizzare l'ID, il nome e la descrizione del gruppo di sicurezza.

7. Scegli la scheda Manutenzione e istantanea per visualizzare i dettagli delle impostazioni delle istantanee.
 - a. In Snapshot, puoi vedere se le istantanee automatizzate sono abilitate, il periodo di conservazione delle istantanee e la finestra delle istantanee.
 - b. In Snapshots, verrà visualizzato un elenco di tutte le istantanee di questo cluster, inclusi il nome, la dimensione, il numero di shard e lo stato delle istantanee.

Per ulteriori informazioni, consulta [Snapshot e ripristino](#).

8. Scegli la scheda Manutenzione e istantanea per visualizzare i dettagli della finestra di manutenzione, insieme a eventuali aggiornamenti ACL, Resharding o Service in sospeso. Per ulteriori informazioni, consulta [Gestione della manutenzione](#).
9. Scegli la scheda Service Updates per visualizzare i dettagli degli eventuali aggiornamenti del servizio applicabili a questo cluster. Per ulteriori informazioni, consulta [Aggiornamenti del servizio in MemoryDB per Redis](#).
10. Scegli la scheda Tag per visualizzare i dettagli di eventuali tag di allocazione delle risorse o dei costi associati a questo cluster. Per ulteriori informazioni, consulta [Taggare le istantanee](#).

Visualizzazione dei dettagli di un cluster (AWSCLI)

Puoi visualizzare i dettagli di un cluster utilizzando il comando dell'AWS CLI `describe-clusters`. Se si omette il parametro `--cluster-name`, vengono restituiti i dettagli relativi a più cluster, fino a un massimo di `--max-results`. Se il parametro `--cluster-name` è incluso, vengono restituiti solo i dettagli relativi al cluster specificato. Puoi limitare il numero di record restituiti con il parametro `--max-results`.

Il codice seguente consente di elencare i dettagli per `my-cluster`.

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

Il codice seguente consente di elencare i dettagli per un massimo di 25 cluster.

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

Example

Per Linux, macOS o Unix:

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

Per Windows:

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

Il seguente output JSON mostra la risposta:

```
{  
  "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-1:0000000000:cluster/my-cluster",
    "SnapshotRetentionLimit": 0,
    "MaintenanceWindow": "sat:06:30-sat:07:30",
    "SnapshotWindow": "04:00-05:00",
    "ACLName": "open-access",
    "DataTiering": "false",
    "AutoMinorVersionUpgrade": true,
}
}

```

Per ulteriori informazioni, consultate l'AWS CLI argomento dedicato a MemoryDB. [describe-clusters](#)

Visualizzazione dei dettagli di un cluster (API MemoryDB)

È possibile visualizzare i dettagli di un cluster utilizzando l'azione API MemoryDB.

DescribeClusters Se il parametro `ClusterName` è incluso, vengono restituiti solo i dettagli relativi al cluster specificato. Se si omette il parametro `ClusterName`, vengono restituiti i dettagli relativi a più cluster, fino a un massimo di `MaxResults` (valore di default 100). Il valore dei `MaxResults` non può essere minore di 20 o maggiore di 100.

Il codice seguente consente di elencare i dettagli per `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>
```

Il codice seguente consente di elencare i dettagli per un massimo di 25 cluster.

```
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>
```

Per ulteriori informazioni, consultate l'argomento di riferimento sull'API MemoryDB.

[DescribeClusters](#)

Modifica di un cluster MemoryDB

Oltre ad aggiungere o rimuovere nodi da un cluster, a volte è necessario apportare altre modifiche a un cluster esistente, ad esempio aggiungere un gruppo di sicurezza, modificare la finestra di manutenzione o un gruppo di parametri.

Consigliamo di impostare la finestra di manutenzione nel momento di utilizzo più basso. Potrebbe essere quindi necessario apportare modifiche di tanto in tanto.

Quando modifichi i parametri di un cluster, la modifica viene applicata al cluster immediatamente. Ciò è valido se modifichi il gruppo dei parametri del cluster o un valore di parametro nel gruppo dei parametri del cluster.

Puoi anche aggiornare la versione del motore dei tuoi cluster. Ad esempio, puoi selezionare una nuova versione secondaria del motore e MemoryDB inizierà immediatamente ad aggiornare il cluster.

Utilizzo di AWS Management Console

Per modificare un cluster

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Dall'elenco nell'angolo in alto a destra, scegli la regione AWS dove si trova il cluster da modificare.
3. Dalla barra di navigazione a sinistra, vai a Clusters. Da Dettagli sui cluster, seleziona il cluster utilizzando il pulsante di opzione e vai su Azioni, quindi su Modifica.
4. Viene visualizzata la pagina Modifica.
5. Nella finestra Modifica, apportate le modifiche desiderate. Le opzioni includono:
 - Description
 - Gruppi di sottoreti
 - Gruppi di sicurezza VPC
 - Tipo di nodo

Note

Se il cluster utilizza un tipo di nodo della famiglia r6gd, è possibile scegliere solo una dimensione del nodo diversa da quella famiglia. Se si sceglie un tipo di nodo dalla

famiglia r6gd, il tiering di dati verrà attivato automaticamente. Per ulteriori informazioni, consulta [Tiering di dati](#).

- Compatibilità della versione Redis
- Abilita le istantanee automatiche
- Periodo di conservazione delle istantanee
- Finestra Snapshot
- Maintenance window (Finestra di manutenzione)
- Argomento per la notifica SNS

6. Seleziona Salvataggio delle modifiche.

Puoi anche andare alla pagina dei dettagli del cluster e fare clic su modifica per apportare modifiche al cluster. Se desideri modificare sezioni specifiche del cluster, puoi andare alla rispettiva scheda nella pagina dei dettagli del cluster e fare clic su Modifica.

Utilizzo di AWS CLI

Puoi modificare un cluster esistente utilizzando l'operazione AWS CLI `update-cluster`. Per modificare il valore di configurazione del cluster, specificare l'ID del cluster, il parametro da modificare e il nuovo valore del parametro. L'esempio seguente modifica la finestra di manutenzione di un cluster denominato `my-cluster` e applica immediatamente la modifica.

Per Linux, macOS o Unix:

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

Per Windows:

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

Per ulteriori informazioni, vedere [update-cluster](#) nel AWS CLI Command Reference.

Utilizzo dell'API MemoryDB

È possibile modificare un cluster esistente utilizzando l'operazione API MemoryDB. [UpdateCluster](#)
Per modificare il valore di configurazione del cluster, specificare l'ID del cluster, il parametro da modificare e il nuovo valore del parametro. L'esempio seguente modifica la finestra di manutenzione di un cluster denominato `my-cluster` e applica immediatamente la modifica.

```
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>
```

Aggiunta/rimozione di nodi da un cluster

È possibile aggiungere o rimuovere nodi da un cluster utilizzando l'AWS Management Console API MemoryDB o AWS CLI l'API MemoryDB.

Utilizzo di AWS Management Console

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Dall'elenco dei cluster, scegli il nome del cluster da cui desideri aggiungere o rimuovere un nodo.
3. Nella scheda Shards and nodes, scegli Aggiungi/Elimina nodi
4. In Nuovo numero di nodi, inserisci il numero di nodi che desideri.
5. Scegli Conferma.

Important

Se imposti il numero di nodi su 1, non sarai più abilitato a Multi-AZ. Puoi anche scegliere di abilitare il failover automatico.

Utilizzo di AWS CLI

1. Identifica i nomi dei nodi che desideri rimuovere. Per ulteriori informazioni, consulta [Visualizzazione dei dettagli di un cluster.](#)
2. Utilizzare l'operazione CLI `update-cluster` con un elenco dei nodi da rimuovere, come nell'esempio seguente.

Per rimuovere nodi da un cluster tramite l'interfaccia a riga di comando, utilizzare il comando `update-cluster` con i seguenti parametri:

- `--cluster-name` L'ID del cluster da cui desideri rimuovere i nodi.
- `--replica-configuration`— Consente di impostare il numero di repliche:
 - `ReplicaCount`— Imposta questa proprietà per specificare il numero di nodi di replica che desideri.
- `--region` Specifica la regione AWS del cluster da cui rimuovere i nodi.

Per Linux, macOS o Unix:

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

Per Windows:

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

Per ulteriori informazioni, consulta gli argomenti AWS CLI [update-cluster](#).

Utilizzo dell'API MemoryDB

Per rimuovere i nodi utilizzando l'API MemoryDB, chiamate l'operazione `UpdateCluster` API con il nome del cluster e un elenco di nodi da rimuovere, come mostrato:

- `ClusterName`— L'ID del cluster da cui desideri rimuovere i nodi.
- `ReplicaConfiguration`— Consente di impostare il numero di repliche:
 - `ReplicaCount`— Imposta questa proprietà per specificare il numero di nodi di replica che desideri.
- `Region`— Specifica la regione AWS del cluster da cui rimuovere un nodo.

Per ulteriori informazioni, vedere [UpdateCluster](#).

Accesso al cluster

Le istanze MemoryDB per Redis sono progettate in modo che vi si possa accedere tramite un'istanza Amazon EC2.

Puoi accedere al nodo MemoryDB da un'istanza Amazon EC2 nello stesso Amazon VPC. Oppure, utilizzando un VPC in peering, puoi accedere al nodo MemoryDB da Amazon EC2 in un diverso Amazon VPC.

Argomenti

- [Concedere l'accesso al cluster](#)
- [Accesso alle risorse MemoryDB dall'esternoAWS](#)

Concedere l'accesso al cluster

Puoi connetterti al cluster MemoryDB solo da un'istanza Amazon EC2 in esecuzione nello stesso Amazon VPC. In questo caso sarà necessario concedere al cluster l'ingresso di rete.

Per concedere l'ingresso di rete a un cluster da un gruppo di sicurezza Amazon VPC

1. Accedere a AWS Management Console e aprire la console Amazon EC2 all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Nel riquadro di navigazione a sinistra, sotto Rete e sicurezza, scegli Security Groups (Gruppi di sicurezza).
3. Dall'elenco dei gruppi di sicurezza, scegliere il gruppo di sicurezza per il VPC Amazon. A meno che il gruppo di sicurezza non sia stato creato per l'utilizzo di MemoryDB, questo gruppo di sicurezza verrà denominato predefinito.
4. Scegliere la scheda Inbound (In entrata), quindi procedere come segue:
 - a. Selezionare Modifica.
 - b. Scegliere Add rule (Aggiungi regola).
 - c. Nella colonna Type (Tipo), scegliere Custom TCP rule (Regola TCP personalizzata).
 - d. Nella casella Port range (Intervallo porta), digitare il numero di porta per il nodo del cluster. Questo numero deve essere lo stesso specificato durante l'avvio del cluster. La porta predefinita per Redis è **6379**.

- e. NellaCreabox, scegliereOvunqueche ha l'intervallo di porta (0.0.0.0/0) in modo che qualunque istanza Amazon EC2 avviata nel VPC Amazon possa connettersi ai nodi MemoryDB.

 Important

Aprire il cluster MemoryDB nell'intervallo di porte 0.0.0.0/0 non espone il cluster a Internet perché sprovvisto di indirizzi IP pubblici e quindi non vi si può accedere dall'esterno del VPC. Tuttavia, il gruppo di sicurezza di default può essere applicato ad altre istanze Amazon EC2 nell'account del cliente, che possono avere un indirizzo IP pubblico. Se eseguono qualche attività sulla porta predefinita, il servizio può essere esposto involontariamente. Consigliamo quindi di creare un gruppo di sicurezza VPC che verrà utilizzato esclusivamente da MemoryDB. Per ulteriori informazioni, consulta [Gruppi di sicurezza personalizzati](#).

- f. Seleziona Save (Salva).

Quando avvii un'istanza Amazon EC2 nel VPC Amazon, questa sarà in grado di connettersi al cluster MemoryDB.

Accesso alle risorse MemoryDB dall'esternoAWS

MemoryDB è un servizio progettato per essere utilizzato internamente al VPC. L'accesso esterno è sconsigliato a causa della latenza del traffico Internet e dei problemi di sicurezza. Tuttavia, se l'accesso esterno a MemoryDB è richiesto per scopi di test o sviluppo, può essere eseguito tramite una VPN.

Utilizzo diAWSClient VPN, consenti l'accesso esterno ai nodi MemoryDB con i seguenti vantaggi:

- Accesso limitato agli utenti approvati o alle chiavi di autenticazione;
- Traffico crittografato tra il client VPN e l'endpoint VPN AWS;
- Accesso limitato a sottoreti o nodi specifici;
- Facile revoca dell'accesso agli utenti o alle chiavi di autenticazione;
- Audit delle connessioni.

Le seguenti procedure dimostrano come:

Argomenti

- [Creare un'autorità di certificazione](#)
- [Configurazione dei componenti AWS VPN Cliente](#)
- [Configurazione del client VPN](#)

Creare un'autorità di certificazione

È possibile creare un'autorità di certificazione (CA) utilizzando tecniche o strumenti diversi. Sugeriamo l'utilità `easy-rsa`, fornita dal progetto [OpenVPN](#). Indipendentemente dall'opzione scelta, assicurati di conservare le chiavi al sicuro. La procedura seguente scarica gli script `easy-rsa`, crea l'autorità di certificazione e le chiavi per autenticare il primo client VPN:

- Per creare i certificati iniziali, apri un terminale e procedi come segue:
 - `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`

Una sottodirectory pki contenente i certificati verrà creata sotto easy-rsa.

- Invia il certificato del server ad AWS Certificate Manager (ACM):
 - Nella console ACM scegliere Certificate Manager (Gestione certificati).
 - scegliere Import Certificate (Importa certificato).
 - Immettere il certificato della chiave pubblica disponibile nel file `easy-rsa/pki/issued/server.crt` nel campo Certificate body (Corpo certificato).
 - Incollare la chiave privata disponibile in `easy-rsa/pki/private/server.key` nel campo Certificate private key (Chiave privata certificato). Assicurarsi di scegliere tutte le righe tra BEGIN AND END PRIVATE KEY (comprese le righe BEGIN ed END).
 - Incollare la chiave pubblica CA disponibile nel file `easy-rsa/pki/ca.crt` nel campo Certificate chain (Catena di certificati).
 - Scegliere Review and import (Verifica e importa).
 - Scegliere Import (Importa).

Per inviare i certificati del server ad ACM utilizzando la CLI AWS, esegui il comando seguente:
`aws acm import-certificate --certificate file://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`

Annota il certificato ARN per uso futuro.

Configurazione dei componenti AWS VPN Cliente

Utilizzo della AWS Console

Nella console AWS scegliere Services (Servizi) e quindi VPC.

In Virtual Private Network (Rete privata virtuale), seleziona Client VPN Endpoints (Endpoint VPN client) ed esegui le operazioni seguenti:

Configurazione dei componenti AWS VPN Cliente

- Seleziona Create Client VPN Endpoint (Crea endpoint VPN client).
- Puoi specificare le seguenti opzioni:

- Client IPv4 CIDR: utilizza una rete privata con una maschera di rete di almeno /22 nel range. Assicurati che la sottorete selezionata non sia in conflitto con gli indirizzi delle reti VPC. Esempio: 10.0.0.0/22.
- In Server certificate ARN (ARN certificato server), seleziona l'ARN del certificato precedentemente importato.
- Seleziona Use mutual authentication (Utilizza autenticazione reciproca).
- In Client certificate ARN (ARN certificato client), seleziona l'ARN del certificato precedentemente importato.
- Seleziona Create Client VPN Endpoint (Crea endpoint VPN client).

Utilizzando la AWS CLI

Eeguire il comando seguente:

```
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
```

Output di esempio:

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

Associazione delle reti di destinazione all'endpoint VPN

- Seleziona il nuovo endpoint VPN, quindi scegliere la scheda Associations (Associazioni).
- Seleziona Associate (Associa) e specifica le seguenti opzioni.
 - VPC: Seleziona il VPC del cluster MemoryDB.
 - Seleziona una delle reti del cluster MemoryDB. In caso di dubbio, esamina le reti nei Gruppi di sottoreti sul pannello di controllo MemoryDB.
 - Seleziona Associate (Associa). Se necessario, ripeti le fasi per le reti rimanenti.

Utilizzando la AWS CLI

Eseguire il comando seguente:

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

Output di esempio:

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

Verifica del gruppo di sicurezza VPN

L'endpoint VPN adotta automaticamente il gruppo di sicurezza di default del VPC. Controlla le regole in ingresso e in uscita e verifica se il gruppo di sicurezza consente il traffico dalla rete VPN (definita nelle impostazioni dell'endpoint VPN) alle reti MemoryDB sulle porte di servizio (per impostazione predefinita, 6379 per Redis).

Se è necessario modificare il gruppo di sicurezza assegnato all'endpoint VPN, procedi come segue:

- Seleziona il gruppo di sicurezza corrente.
- Seleziona Apply Security Group (Applica gruppo di sicurezza).
- Scegliere il nuovo gruppo di sicurezza.

Utilizzando la AWS CLI

Eseguire il comando seguente:

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

Output di esempio:

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

Note

Il gruppo di sicurezza MemoryDB deve inoltre consentire il traffico proveniente dai client VPN. Gli indirizzi dei client saranno mascherati con l'indirizzo dell'endpoint VPN, in base alla

rete VPC. Pertanto, considera la rete VPC (non la rete client VPN) quando crei la regola in ingresso nel gruppo di sicurezza MemoryDB.

Autorizzazione dell'accesso VPN alle reti di destinazione

Nella scheda Authorization (Autorizzazione) seleziona Authorize Ingress (Autorizza ingresso) e specifica quanto segue:

- Rete di destinazione per abilitare l'accesso: Utilizzare 0.0.0.0/0 per consentire l'accesso a qualsiasi rete (incluso Internet) o limitare le reti/gli host MemoryDB.
- In Grant access to: (Concedi accesso a:), seleziona Allow access to all users (Consenti accesso a tutti gli utenti).
- Seleziona Add Authorization Rules (Aggiungi regole di autorizzazione).

Utilizzando la AWS CLI

Eeguire il comando seguente:

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

Output di esempio:

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

Autorizzazione dell'accesso a Internet dai client VPN

Se è necessario navigare in Internet tramite la VPN, dovrai creare una route aggiuntiva. Seleziona la scheda Route Table (Tabella di routing) e quindi seleziona Create Route (Crea route):

- Destinazione del percorso: 0.0.0.0/0
- ID subnet VPC di destinazione: Seleziona una delle sottoreti associate con accesso a Internet.
- Seleziona Create Route (Crea route).

Utilizzando la AWS CLI

Eeguire il comando seguente:

```
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
```

Output di esempio:

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

Configurazione del client VPN

Nel pannello di controllo VPN Cliente AWS seleziona l'endpoint VPN creato di recente e seleziona Download Client Configuration (Scarica configurazione client). Copia il file di configurazione e i file `easy-rsa/pki/issued/client1.domain.tld.crt` e `easy-rsa/pki/private/client1.domain.tld.key`. Modifica il file di configurazione e modifica o aggiungi i seguenti parametri:

- `cert`: aggiungi una nuova riga con il parametro `cert` che punta al file `client1.domain.tld.crt`. Usa il percorso completo del file. Esempio: `cert /home/user/.cert/client1.domain.tld.crt`
- `cert: key`: aggiungi una nuova riga con la chiave del parametro che punta al file `client1.domain.tld.key`. Usa il percorso completo del file. Esempio: `key /home/user/.cert/client1.domain.tld.key`

Stabilisci la connessione VPN con il comando: `sudo openvpn --config downloaded-client-config.ovpn`

Revoca dell'accesso

Se è necessario invalidare l'accesso a una particolare chiave client, la chiave deve essere revocata nella CA. Invia quindi l'elenco di revoca a AWS VPN Cliente.

Revoca la chiave con `easy-rsa`:

- `cd easy-rsa`
- `./easyrsa3/easyrsa revoke client1.domain.tld`
- Inserisci "sì" per continuare o qualsiasi altro input per interrompere.

```
Continue with revocation: `yes` ... * `./easyrsa3/easyrsa gen-crl
```

- È stato creato un file CRL aggiornato. File CRL: `/home/user/easy-rsa/pki/crl.pem`

Importazione dell'elenco di revoca in AWS VPN Cliente:

- Nella AWS Management Console, seleziona Services (Servizi) e quindi VPC.
- Seleziona Client VPN Endpoints (Endpoint client VPN).
- Seleziona l'endpoint VPN client e quindi seleziona Actions (Operazioni) -> Import Client Certificate CRL (Importa CRL certificato client).
- Incolla il contenuto del file `crl.pem`:

Utilizzando la AWS CLI

Eseguire il comando seguente:

```
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
```

Output di esempio:

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

Individuazione degli endpoint di connessione

La tua applicazione si connette al cluster utilizzando l'endpoint. Un endpoint corrisponde all'indirizzo univoco di un cluster. Usa i `clusterEndpoint` del cluster per tutte le operazioni

Le sezioni seguenti consentono di individuare l'endpoint di cui si ha bisogno.

Ricerca dell'endpoint per un cluster MemoryDB (AWS Management Console)

Per individuare l'endpoint di un cluster MemoryDB

1. Accedi allaAWS Management Consolee aprire la console MemoryDB per Redis all'indirizzo<https://console.aws.amazon.com/memorydb/>.
2. Nel riquadro di navigazione scegliere Clusters (Cluster).

Verrà visualizzata la schermata con l'elenco dei cluster. Scegli il cluster a cui desideri connetterti.

3. Per trovare l'endpoint del cluster, scegli il nome del cluster (non il pulsante di opzione).
4. L'Endpoint del clusterè visualizzato sottoDettagli del cluster. Per copiarlo, scegli copiaicona a sinistra dell'endpoint.

Ricerca dell'endpoint per un cluster MemoryDB (AWSCLI)

Puoi utilizzare il plugindescribe-clusterscomando per trovare l'endpoint per un cluster. Il comando restituisce l'endpoint del cluster.

L'operazione seguente recupera l'endpoint, che in questo esempio è rappresentato come*provare*, per il clustermycluster.

Restituisce la risposta JSON seguente:

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

Per 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  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [describe-clusters](#).

Ricerca

Puoi trovare AWS Management Console AWS CLI il nome

Utilizzo di AWS Management Console

La procedura seguente utilizza il AWS Management Console per trovare i nomi degli shard del cluster di MemoryDB.

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Nel pannello di navigazione sulla sinistra, scegli Clupi.
3. Scegli il cluster sotto Nome di cui vuoi trovare i nomi degli shard.
4. Nella scheda Frammenti e nodi, visualizza l'elenco degli shard sotto Nome. Puoi anche espandere ognuno di essi per visualizzare i dettagli dei relativi nodi.

Utilizzo di AWS CLI

Per trovare i nomi shard (shard) per i cluster MemoryDB, utilizzare l'AWS CLI operazione `describe-clusters` con il seguente parametro opzionale.

- **--cluster-name**- Parametro facoltativo che, se utilizzato, limita l'output ai dettagli del cluster Se questo parametro viene omesso, vengono omesso, vengono omesso, vengono omesso,
- **--show-shard-details**—Restituisce i dettagli dei frammenti, inclusi i loro nomi.

Questo comando restituisce i dettagli per `my-cluster`.

Per Linux, macOS o Unix:

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

Per Windows:

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

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

Restituisce la risposta JSON

Le interruzioni di riga vengono aggiunte per facilitare la lettura.

```
{
  "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
  }
]
}

```

Utilizzo

Per trovare gli ID degli shard per i cluster MemoryDB, utilizza l'operazione API `DescribeClusters` con il seguente parametro opzionale.

- **ClusterName**— Parametro facoltativo che, se utilizzato, limita l'output ai dettagli del cluster. Se questo parametro viene omesso, vengono omesso, vengono omesso, vengono omesso.
- **ShowShardDetails**— Restituisce i dettagli dei frammenti, inclusi i loro nomi.

Example

Questo comando restituisce i dettagli per `my-cluster`.

Per 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>
```

Gestione dell'implementazione di MemoryDB

In questa sezione, puoi trovare dettagli su come gestire i vari componenti dell'implementazione di MemoryDB.

Argomenti

- [Versioni del motore Redis](#)
- [Nozioni di base di JSON](#)
- [Tagging delle risorse MemoryDB](#)
- [Gestione della manutenzione](#)
- [Best practice](#)
- [Informazioni sulla replica MemoryDB](#)
- [Snapshot e ripristino](#)
- [Dimensionamento](#)
- [Configurazione dei parametri di motore con i gruppi di parametri](#)
- [Tutorial: Configurazione di una funzione Lambda per accedere a MemoryDB in un Amazon VPC](#)

Versioni del motore Redis

Questa sezione copre le versioni supportate del motore Redis.

Argomenti

- [MemoryDB per Redis versione 7.1 \(migliorata\)](#)
- [MemoryDB per Redis versione 7.0 \(migliorata\)](#)
- [MemoryDB per Redis versione 6.2 \(migliorata\)](#)
- [Aggiornamento delle versioni del motore](#)

MemoryDB per Redis versione 7.1 (migliorata)

MemoryDB for Redis versione 7.1 aggiunge il supporto per le funzionalità di ricerca vettoriale in anteprima per regioni selezionate, oltre a correzioni di bug critici e miglioramenti delle prestazioni.

- Funzione di ricerca [vettoriale](#): [la ricerca vettoriale può essere utilizzata con le funzionalità MemoryDB esistenti](#). Le applicazioni che non utilizzano la ricerca vettoriale non saranno influenzate dalla sua presenza. L'anteprima della ricerca vettoriale è disponibile in MemoryDB per Redis dalla versione 7.1 in poi nelle seguenti regioni: Stati Uniti orientali (Virginia settentrionale e Ohio), Stati Uniti occidentali (Oregon), UE (Irlanda) e Asia Pacifico (Tokyo). Consulta la documentazione [qui](#) per sapere come abilitare l'anteprima della ricerca vettoriale e le funzionalità correlate.

Note

MemoryDB per Redis versione 7.1 è compatibile con OSS Redis v7.0. [Per ulteriori informazioni sulla versione Redis 7.0, consulta le note di rilascio di Redis 7.0 su Redis on GitHub](#)

MemoryDB per Redis versione 7.0 (migliorata)

MemoryDB per Redis 7.0 aggiunge una serie di miglioramenti e supporto per nuove funzionalità:

- [Funzioni Redis](#): MemoryDB per Redis 7 aggiunge il supporto per Redis Functions e offre un'esperienza gestita che consente agli sviluppatori di eseguire [script LUA con la logica dell'applicazione archiviata nel cluster MemoryDB, senza richiedere ai client di inviare nuovamente gli script](#) al server ad ogni connessione.
- [Miglioramenti ACL](#): MemoryDB per Redis 7 aggiunge il supporto per la prossima versione di Redis Access Control Lists (ACL). Con MemoryDB per Redis 7, i client possono ora specificare più set di autorizzazioni su chiavi o spazi chiave specifici in Redis.
- [Sharded Pub/Sub](#): MemoryDB for Redis 7 aggiunge il supporto per l'esecuzione della funzionalità Redis Pub/Sub in modo condiviso quando si esegue MemoryDB in Cluster Mode Enabled (CME). Le funzionalità Redis Pub/Sub consentono agli editori di inviare messaggi a qualsiasi numero di sottoscrittori su un canale. Con Amazon MemoryDB per Redis 7, i canali sono associati a uno shard nel cluster MemoryDB, eliminando la necessità di propagare le informazioni del canale tra gli shard. Ciò si traduce in una migliore scalabilità.
- [Multiplexing I/O migliorato](#): MemoryDB per Redis versione 7 introduce il multiplexing I/O avanzato, che offre un throughput maggiore e una latenza ridotta per carichi di lavoro ad alto throughput con molte connessioni client simultanee a un cluster MemoryDB. Ad esempio, quando si utilizza un cluster di nodi r6g.4xlarge e si eseguono 5200 client simultanei, è possibile ottenere un aumento

del throughput fino al 46% (operazioni di lettura e scrittura al secondo) e una riduzione della latenza P99 fino al 21%, rispetto a MemoryDB per Redis versione 6.

[Per ulteriori informazioni sulla versione Redis 7.0, consulta le note di rilascio di Redis 7.0 su Redis on. GitHub](#)

MemoryDB per Redis versione 6.2 (migliorata)

MemoryDB introduce la prossima versione del motore Redis, che include il supporto per l'aggiornamento automatico della versione, la memorizzazione nella [Autenticazione degli utenti con elenchi di controllo degli accessi \(ACL\)](#) cache lato client e significativi miglioramenti operativi.

La versione 6.2.6 del motore Redis introduce anche il supporto per il formato nativo di JavaScript Object Notation (JSON), un modo semplice e senza schemi per codificare set di dati complessi all'interno dei cluster Redis. Con il supporto JSON, puoi sfruttare le prestazioni e le API Redis per applicazioni che operano su JSON. Per ulteriori informazioni, consulta [Nozioni di base di JSON](#). È inclusa anche la metrica relativa a JSON incorporata per monitorare l'utilizzo di questo tipo di dati. `JsonBasedCmds` CloudWatch Per ulteriori informazioni, consulta [Metriche per MemoryDB](#).

Con Redis 6, MemoryDB offrirà un'unica versione per ogni versione minore di Redis OSS, anziché offrire più versioni di patch. Questo è progettato per ridurre al minimo la confusione e l'ambiguità derivanti dalla necessità di scegliere tra più versioni minori. MemoryDB gestirà inoltre automaticamente la versione secondaria e la versione patch dei cluster in esecuzione, garantendo prestazioni migliori e maggiore sicurezza. Ciò verrà gestito tramite canali standard di notifica ai clienti tramite una campagna di aggiornamento del servizio. Per ulteriori informazioni, consulta [Aggiornamenti del servizio in MemoryDB per Redis](#).

Se non specificate la versione del motore durante la creazione, MemoryDB selezionerà automaticamente la versione Redis preferita. D'altra parte, se si specifica la versione del motore utilizzando `6.2`, MemoryDB richiamerà automaticamente la versione patch preferita di Redis 6.2 disponibile.

Ad esempio, quando si crea un cluster, si imposta il parametro su `--engine-version 6.2` Il cluster verrà avviato con la versione di patch preferita attualmente disponibile al momento della creazione. Qualsiasi richiesta con un valore di versione completa del motore verrà rifiutata, verrà generata un'eccezione e il processo avrà esito negativo.

Quando si chiama l'`DescribeEngineVersionsAPI`, il valore del `EngineVersion` parametro verrà impostato su 6.2 e la versione completa effettiva del motore verrà restituita nel `EnginePatchVersion` campo.

Per ulteriori informazioni sulla versione Redis 6.2, consulta le note di rilascio di [Redis 6.2](#) su [Redis on GitHub](#).

Aggiornamento delle versioni del motore

Per impostazione predefinita, MemoryDB gestisce automaticamente la versione patch dei cluster in esecuzione tramite gli aggiornamenti del servizio. Puoi inoltre disattivare l'aggiornamento automatico della versione secondaria se imposti la `AutoMinorVersionUpgrade` proprietà dei tuoi cluster su `false`. Tuttavia, non è possibile disattivare l'aggiornamento automatico della versione della patch.

È possibile controllare se e quando il software conforme al protocollo che alimenta il cluster viene aggiornato alle nuove versioni supportate da MemoryDB prima dell'avvio dell'aggiornamento automatico. Questo livello di controllo ti consente di mantenere la compatibilità con versioni specifiche, testare le nuove versioni con l'applicazione prima di distribuirle in produzione e aggiornare le versioni alle tue condizioni e secondo le tue scadenze.

È possibile avviare gli aggiornamenti della versione del motore del cluster nei seguenti modi:

- Aggiornandolo e specificando una nuova versione del motore. Per ulteriori informazioni, consulta [Modifica di un cluster MemoryDB](#).
- Applicazione dell'aggiornamento del servizio per la versione del motore corrispondente. Per ulteriori informazioni, consulta [Aggiornamenti del servizio in MemoryDB per Redis](#).

Tieni presente quanto segue:

- Puoi eseguire l'aggiornamento a una versione del motore più recente; non è consentito, invece, il downgrade a versioni precedenti. Se vuoi utilizzare una versione del motore precedente, elimina il cluster esistente e crealo di nuovo con la versione del motore precedente.
- È preferibile eseguire periodicamente l'aggiornamento all'ultima versione principale, siccome la maggior parte dei miglioramenti principali non viene ripristinata alle versioni precedenti. Man mano che MemoryDB espande la disponibilità in una nuova AWS regione, MemoryDB supporta le due `MAJOR.MINOR` versioni più recenti in quel momento per la nuova regione. Ad esempio, se viene avviata una nuova AWS regione e le versioni più recenti di `MAJOR.MINOR` MemoryDB per Redis sono 7.0 e 6.2, MemoryDB for Redis supporterà le versioni 7.0 e 6.2 nella nuova regione. AWS

Man mano che verranno MAJOR .MINOR rilasciate le versioni più recenti di MemoryDB per Redis, MemoryDB continuerà ad aggiungere il supporto per la nuova versione di MemoryDB for Redis. Per ulteriori informazioni sulla scelta delle regioni per MemoryDB, consulta [Regioni ed endpoint supportati](#)

- La gestione della versione del motore è progettata in modo da avere il maggior controllo possibile sulle modalità di applicazione delle patch. Tuttavia, MemoryDB si riserva il diritto di applicare patch al cluster per conto dell'utente nell'improbabile eventualità che si verifichi una vulnerabilità critica di sicurezza nel sistema o nel software.
- MemoryDB offrirà un'unica versione per ogni versione minore di Redis OSS, anziché offrire più versioni di patch. Questo è progettato per ridurre al minimo la confusione e l'ambiguità derivanti dalla necessità di scegliere tra più versioni. MemoryDB gestirà inoltre automaticamente la versione secondaria e la versione patch dei cluster in esecuzione, garantendo prestazioni migliorate e maggiore sicurezza. Ciò verrà gestito tramite canali standard di notifica ai clienti tramite una campagna di aggiornamento del servizio. Per ulteriori informazioni, consulta [Aggiornamenti del servizio in MemoryDB per Redis](#).
- È possibile aggiornare la versione del cluster con tempi di inattività minimi. Il cluster è disponibile per la lettura durante l'intero aggiornamento ed è disponibile per la scrittura durante la maggior parte della sua durata, eccetto durante l'operazione di failover che dura alcuni secondi.
- Ti consigliamo di eseguire gli aggiornamenti del motore durante i periodi di basso traffico di scrittura in entrata.

I cluster con più shard vengono elaborati e patchati come segue:

- Viene eseguita una sola operazione di aggiornamento per shard alla volta.
- In ogni partizione, tutte le repliche vengono elaborate prima del primario. Se una partizione annovera poche repliche, il suo nodo primario potrebbe giungere alla conclusione dell'elaborazione prima delle repliche negli altre partizioni.
- I nodi primari delle varie partizioni vengono elaborati in serie. Viene aggiornato un solo nodo primario alla volta.

Argomenti

- [Come aggiornare la versione di un motore](#)
- [Risolvere il blocco di un aggiornamento del motore Redis](#)

Come aggiornare la versione di un motore

È possibile avviare gli aggiornamenti di versione del cluster modificandolo utilizzando la console MemoryDB, l'API MemoryDB o l'API MemoryDB e AWS CLI specificando una versione più recente del motore. Per ulteriori informazioni, consulta i seguenti argomenti.

- [Utilizzo di AWS Management Console](#)
- [Utilizzo di AWS CLI](#)
- [Utilizzo dell'API MemoryDB](#)

Risolvere il blocco di un aggiornamento del motore Redis

Come illustrato nella tabella seguente, l'aggiornamento di un motore Redis si blocca in presenza di operazioni di dimensionamento in sospeso

Operazioni in sospeso	Operazioni bloccate
Dimensionamento	Aggiornamento immediato del motore
Aggiornamento del motore	Dimensionamento immediato
Dimensionamento e aggiornamento del motore	Dimensionamento immediato
	Aggiornamento immediato del motore

Nozioni di base di JSON

MemoryDB supporta il formato JSON (JavaScriptObject Notation) nativo, un modo semplice e senza schemi per codificare set di dati complessi nei cluster Redis. Puoi archiviare e accedere in modo nativo ai dati utilizzando il formato JavaScript Object Notation (JSON) nei cluster Redis e aggiornare i dati JSON memorizzati in tali cluster senza la necessità di gestire codice personalizzato per serializzarlo e deserializzarlo.

Oltre a sfruttare le API Redis per le applicazioni che funzionano su JSON, ora puoi recuperare e aggiornare in modo efficiente parti specifiche di un documento JSON senza dover manipolare l'intero oggetto, il che può migliorare le prestazioni e ridurre i costi. Puoi cercare anche contenuti del documento JSON utilizzando l'interrogazione JSONPath di [tipo Goessner](#).

Dopo aver creato un cluster con una versione del motore supportata, il tipo di dati JSON e i comandi associati sono automaticamente disponibili. È compatibile con le API e RDB con la versione 2 del modulo RedisJSON, quindi puoi migrare facilmente le applicazioni Redis basate su JSON esistenti in MemoryDB. Per ulteriori informazioni sui comandi Redis supportati, consulta [Comandi supportati](#).

La metrica relativa a JSON `JsonBasedCmds` è incorporata CloudWatch per monitorare l'utilizzo di questo tipo di dati. Per ulteriori informazioni, consulta [Metrics for MemoryDB](#).

Note

Per utilizzare JSON, devi eseguire il motore Redis versione 6.2.6 o successiva.

Argomenti

- [Panoramica del tipo di dati Redis JSON](#)
- [Comandi supportati](#)

Panoramica del tipo di dati Redis JSON

MemoryDB supporta una serie di comandi Redis per lavorare con il tipo di dati JSON. Di seguito è riportata una panoramica del tipo di dati JSON e un elenco dettagliato dei comandi Redis supportati.

Terminology

Termine	Descrizione
Documento JSON	si riferisce al valore di una chiave Redis JSON
Valore JSON	fa riferimento a un sottoinsieme di un documento JSON, inclusa la radice che rappresenta l'intero documento. Un valore può essere un contenitore o una voce all'interno di un contenitore
Elemento JSON	equivalente al valore JSON

Standard JSON supportati

Il formato JSON è compatibile con lo standard di interscambio dati JSON [RFC 7159](#) e [ECMA-404](#). Nel testo JSON è supportato [Unicode](#) UTF-8.

Elemento radice

L'elemento radice può essere qualsiasi tipo di dati JSON. Tieni presente che nello standard RFC 4627 precedente, come valori radice erano consentiti solo oggetti o array. Dopo l'aggiornamento allo standard RFC 7159, la radice di un documento JSON può essere qualunque tipo di dati JSON.

Limite delle dimensioni del documento

I documenti JSON sono archiviati internamente in un formato ottimizzato per un accesso e una modifica rapidi. Questo formato comporta in genere un consumo leggermente maggiore di memoria rispetto alla rappresentazione serializzata equivalente dello stesso documento. Il consumo di memoria di un singolo documento JSON è limitato a 64 MB, cioè la dimensione della struttura dati in memoria, non la stringa JSON. La quantità di memoria consumata da un documento JSON può essere controllata utilizzando il `JSON.DEBUG MEMORY` comando.

ACL JSON

- Il tipo di dati JSON è completamente integrato nella funzionalità [Redis Access Control Lists](#) (ACL). Analogamente alle categorie esistenti per tipo di dati (`@string`, `@hash`, ecc.), è stata aggiunta una nuova categoria `@json` per semplificare la gestione dell'accesso ai comandi e ai dati JSON. Nessun altro comando Redis esistente è membro della categoria `@json`. Tutti i comandi JSON impongono restrizioni e autorizzazioni per lo spazio delle chiavi o i comandi.
- Esistono cinque categorie di ACL Redis esistenti che vengono aggiornate per includere i nuovi comandi JSON: `@read`, `@write`, `@fast`, `@slow` e `@admin`. La tabella seguente indica la mappatura dei comandi JSON alle categorie appropriate.

ACL

Comando JSON	@read	@write	@fast	@slow	@admin
JSON.ARRAPPEND		y	y		

Comando JSON	@read	@write	@fast	@slow	@admin
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		
JSON.NUMMULTBY		y	y		

Comando JSON	@read	@write	@fast	@slow	@admin
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		

Limite di profondità di nidificazione

Quando un oggetto o un array JSON ha un elemento che è esso stesso un altro oggetto o array JSON, si dice che tale oggetto o array si nidifica nell'oggetto o nell'array esterno. Il limite massimo di profondità di nidificazione è 128. Qualunque tentativo di creare un documento che contenga una profondità di nidificazione maggiore di 128 verrà rifiutato con un errore.

Sintassi dei comandi

La maggior parte dei comandi richiede un nome chiave Redis come primo argomento. Alcuni comandi hanno anche un argomento path. Se non è fornito, viene ripristinato il percorso predefinito, la radice.

Notazione:

- Gli argomenti obbligatori sono racchiusi tra parentesi angolari, ad es. <key>
- Gli argomenti opzionali sono racchiusi tra parentesi quadre, ad esempio [percorso]
- Gli argomenti opzionali aggiuntivi sono indicati da..., ad esempio [json...]

Sintassi del percorso

JSON-Redis supporta due tipi di sintassi dei percorsi:

- Sintassi avanzata: segue la sintassi JSONPath descritta da [Goessner](#), come mostrato nella tabella seguente. Abbiamo riordinato e modificato le descrizioni nella tabella per maggiore chiarezza.
- Sintassi limitata: ha limitate capacità di interrogazione.

Note

I risultati di alcuni comandi dipendono dal tipo di sintassi del percorso utilizzato.

Se un percorso di interrogazione inizia con '\$', utilizza la sintassi avanzata. In caso contrario, viene utilizzata la sintassi limitata.

Sintassi migliorata

Simbolo/espressione	Descrizione
\$	l'elemento radice
. o []	operatore bambino
..	discesa ricorsiva

Simbolo/espressione	Descrizione
*	jolly. Tutti gli elementi di un oggetto o un array.
[]	operatore di pedice dell'array. L'indice è basato su 0.
[,]	operatore sindacale
[start:end:step]	operatore di blocco dell'array
?()	applica un'espressione di filtro (script) all'array o all'oggetto corrente
()	espressione di filtro
@	utilizzato nelle espressioni di filtro che si riferiscono al nodo corrente in fase di elaborazione
==	uguale a, utilizzato nelle espressioni di filtro.
!=	diverso da, utilizzato nelle espressioni di filtro.
>	maggiore di, utilizzato nelle espressioni di filtro.
>=	maggiore o uguale a, utilizzato nelle espressioni di filtro.
<	inferiore a, utilizzato nelle espressioni di filtro.
<=	minore o uguale a, utilizzato nelle espressioni di filtro.
&&	AND logico, utilizzato per combinare più espressioni di filtro.
	OR logico, utilizzato per combinare più espressioni di filtro.

Examples (Esempi)

Gli esempi seguenti sono basati sui dati XML [di esempio di Goessner](#), che abbiamo modificato aggiungendo campi aggiuntivi.

```
{ "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
  }
}
```

```
}
}
```

Percorso	Descrizione
<code>\$.store.book[*].author</code>	gli autori di tutti i libri presenti nel negozio
<code>\$..author</code>	tutti gli autori
<code>\$.store.*</code>	tutti i membri del negozio
<code>\$["store"].*</code>	tutti i membri del negozio
<code>\$.store..price</code>	il prezzo di tutto ciò che c'è nel negozio
<code>\$..*</code>	tutti i membri ricorsivi della struttura JSON
<code>\$.book[*]</code>	tutti i libri
<code>\$.book[0]</code>	il primo libro
<code>\$.book[-1]</code>	l'ultimo libro
<code>\$.book[0:2]</code>	i primi due libri
<code>\$.book[0,1]</code>	i primi due libri
<code>\$.book[0:4]</code>	libri dall'indice 0 a 3 (l'indice finale non è incluso)
<code>\$.book[0:4:2]</code>	libri all'indice 0, 2
<code>\$.book[?(@.isbn)]</code>	tutti i libri con numero ISBN
<code>\$.book[?(@.price<10)]</code>	tutti i libri a un prezzo inferiore a \$10
<code>'\$.book[?(@.price < 10)]'</code>	tutti i libri a un prezzo inferiore a \$10. (Il percorso deve essere citato se contiene spazi bianchi)
<code>'\$.book[?(@["price"] < 10)]'</code>	tutti i libri a un prezzo inferiore a \$10

Percorso	Descrizione
'\$.book[?(@.["price"] < 10)]'	tutti i libri a un prezzo inferiore a \$10
\$.book[?(@.price>=10&&@.price<=100)]	tutti i libri nella fascia di prezzo compresa tra \$10 e \$100, inclusi
'\$.book[?(@.price>=10 && @.price<=100)]'	tutti i libri nella fascia di prezzo compresa tra \$10 e \$100, inclusi. (Il percorso deve essere citato se contiene spazi bianchi)
\$.book[?(@.sold==true @.in-stock==false)]	tutti i libri venduti o esauriti
'\$.book[?(@.sold == true @.in-stock == false)]'	tutti i libri venduti o esauriti. (Il percorso deve essere citato se contiene spazi bianchi)
'\$.store.book[?(@.["category"] == "fiction")]'	tutti i libri della categoria narrativa
'\$.store.book[?(@.["category"] != "fiction")]'	tutti i libri nelle categorie di saggistica

Altri esempi di espressioni di 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]"

```

Sintassi limitata

Simbolo/espressione	Descrizione
. o []	operatore bambino
[]	operatore di pedice dell'array. L'indice è basato su 0.

Examples (Esempi)

Percorso	Descrizione
.store.book[0].author	l'autore del primo libro
.store.book[-1].author	l'autore dell'ultimo libro
.address.city	nome della città
["store"]["book"][0]["title"]	il titolo del primo libro
["store"]["book"][-1]["title"]	il titolo dell'ultimo libro

Note

Tutti i contenuti di [Goessner](#) citati in questa documentazione sono soggetti alla [Creative Commons License](#).

Prefissi di errori comuni

Ogni messaggio di errore ha un prefisso. Di seguito è riportato un elenco di prefissi di errore comuni:

Prefix	Descrizione
ERR	un errore generale
LIMIT	errore limite di dimensione superato. Ad esempio, il limite di dimensione del documento o il limite di profondità di nidificazione sono stati superati
NONEXISTENT	una chiave o un percorso non esiste
OUTOFBOUNDARIES	indice dell'array fuori dai limiti
SYNTAXERR	errore di sintassi
WRONGTYPE	tipo di valore errato

Metriche correlate a JSON

Di seguito sono fornite le seguenti metriche di informazioni JSON:

Info	Descrizione
json_total_memory_bytes	memoria totale allocata agli oggetti JSON
json_num_documents	numero totale di documenti in Redis

Per interrogare le metriche di base, esegui il comando Redis:

```
info json_core_metrics
```

Come MemoryDB interagisce con JSON

Di seguito viene illustrato come MemoryDB interagisce con il tipo di dati JSON.

Precedenza degli operatori

Durante la valutazione delle espressioni condizionali per il filtro, `&&` hanno la precedenza, quindi vengono valutati `||`, come nella maggior parte dei linguaggi. Le operazioni all'interno delle parentesi verranno eseguite per prime.

Comportamento del limite massimo di nidificazione dei percorsi

Il limite massimo di nidificazione dei percorsi di MemoryDB è 128. Per cui, un valore come `$.a.b.c.d...` può raggiungere solo 128 livelli.

Gestione dei valori numerici

JSON non dispone di tipi di dati separati per numeri interi e numeri in virgola mobile. Sono tutti definiti "numeri".

Quando viene ricevuto un numero JSON, questo viene memorizzato in uno dei due formati. Se il numero rientra in un numero intero con segno a 64 bit, viene convertito in quel formato; in caso contrario, viene memorizzato come stringa. Le operazioni aritmetiche su due numeri JSON (ad esempio `JSON.NUMINCRBY` e `JSON.NUMMULTBY`) cercano di preservare la massima precisione possibile. Se i due operandi e il valore risultante si inseriscono in un numero intero con segno a 64 bit, viene eseguita l'aritmetica dei numeri interi. Altrimenti, gli operandi di input vengono convertiti in numeri a virgola mobile IEEE a doppia precisione IEEE a 64 bit, viene eseguita l'operazione aritmetica e il risultato viene riconvertito in una stringa.

Comandi aritmetici `JSON.NUMINCRBY` e `JSON.NUMMULTBY`:

- Se entrambi i numeri sono numeri interi e il risultato non rientra nell'intervallo di `int64`, diventerà automaticamente un numero a virgola mobile a doppia precisione.
- Se almeno uno dei numeri è in virgola mobile, il risultato sarà un numero a virgola mobile a doppia precisione.
- Se il risultato supera l'intervallo di `double`, il comando restituirà un `OVERFLOW` errore.

Note

Prima della versione 6.2.6.R2 del motore Redis, quando viene ricevuto un numero JSON in ingresso, questo viene convertito in una delle due rappresentazioni binarie interne: un numero intero con segno a 64 bit o una virgola mobile IEEE a doppia precisione a 64 bit. La stringa originaria e tutta la formattazione non vengono mantenute. Pertanto,

quando un numero viene emesso come parte di una risposta JSON, viene convertito dalla rappresentazione binaria interna in una stringa stampabile che utilizza regole di formattazione generiche. Queste regole potrebbero determinare la generazione di una stringa diversa da quella ricevuta.

- Se entrambi i numeri sono numeri interi e il risultato non rientra nell'intervallo di `int64`, diventa automaticamente un numero a virgola mobile a doppia precisione IEEE a 64 bit.
- Se almeno uno dei numeri è in virgola mobile, il risultato è un numero in virgola mobile a doppia precisione IEEE a 64 bit.
- Se il risultato supera l'intervallo del doppio IEEE a 64 bit, il comando restituisce un `OVERFLOW` errore.

Per un elenco dei comandi disponibili, consulta [Comandi supportati](#).

Valutazione della sintassi rigida

MemoryDB non consente percorsi JSON con sintassi non valida, neppure se un sottoinsieme del percorso contiene un percorso valido. Ciò per mantenere un comportamento corretto per i nostri clienti.

Comandi supportati

Sono supportati i seguenti comandi Redis:

Argomenti

- [JSON.ARRAPPEND](#)
- [JSON.ARRINDEX](#)
- [JSON.ARRINSERT](#)
- [JSON.ARRLEN](#)
- [JSON.ARRPOP](#)
- [JSON.ARRTRIM](#)
- [JSON.CLEAR](#)
- [JSON.DEBUG](#)
- [JSON.DEL](#)
- [JSON.FORGET](#)

- [JSON.GET](#)
- [JSON.MGET](#)
- [JSON.NUMINCRBY](#)
- [JSON.NUMMULTBY](#)
- [JSON.OBJLEN](#)
- [JSON.OBJKEYS](#)
- [JSON.RESP](#)
- [JSON.SET](#)
- [JSON.STRAPPEND](#)
- [JSON.STRLEN](#)
- [JSON.TOGGLE](#)
- [JSON.TYPE](#)

JSON.ARRAPPEND

Aggiunge uno o più valori ai valori dell'array nel percorso.

Sintassi

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

- **key** (richiesto) — chiave Redis di tipo documento JSON
- **percorso** (obbligatorio) — un percorso JSON
- **json** (obbligatorio) — Valore JSON da aggiungere all'array

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di numeri interi, che rappresenta la nuova lunghezza dell'array in ogni percorso.
- Se un valore non è un array, il valore restituito corrispondente è null.
- Errore SYNTAXERR se uno degli argomenti json di input non è una stringa JSON valida.
- Errore NONEXISTENT se il percorso non esiste.

Se il percorso è una sintassi limitata:

- Numero intero, la nuova lunghezza dell'array.
- Se sono selezionati più valori array, il comando restituisce la nuova lunghezza dell'ultimo array aggiornato.
- Errore WRONGTYPE se il valore nel percorso non è un array.
- Errore SYNTAXERR se uno degli argomenti json di input non è una stringa JSON valida.
- Errore NONEXISTENT se il percorso non esiste.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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\"]]"
```

Sintassi limitata del percorso:

```
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

Cerca la prima occorrenza di un valore JSON scalare negli array del percorso.

- Gli errori “fuori intervallo” vengono gestiti arrotondando l'indice all'inizio e alla fine dell'array.
- Se l'inizio è maggiore della fine, restituisce -1 (non trovato).

Sintassi

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

- **key** (richiesto) — chiave Redis di tipo documento JSON
- **percorso** (obbligatorio) — un percorso JSON
- **json-scalar** (obbligatorio) — valore scalare da cercare; JSON scalar si riferisce a valori che non sono oggetti o array. Ad esempio, String, number, boolean e null sono valori scalari.
- **start** (opzionale) — indice iniziale, incluso. Se non è fornito, viene utilizzata l'impostazione predefinita, 0.
- **fine** (opzionale) — indice finale, esclusivo. Viene ripristinato il valore predefinito 0 se non è fornito, ossia l'ultimo elemento. 0 o -1 significa che è incluso l'ultimo elemento.

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di numeri interi. Ogni valore è l'indice dell'elemento corrispondente nell'array nel percorso. Se non viene trovato, il valore è -1.
- Se un valore non è un array, il valore restituito corrispondente è null.

Se il percorso è una sintassi limitata:

- Numero intero, l'indice dell'elemento corrispondente o -1 se non viene trovato.
- Errore `WRONGTYPE` se il valore nel percorso non è un array.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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
```

Sintassi limitata del percorso:

```
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

Inserisci uno o più valori nei valori dell'array nel percorso prima dell'indice.

Sintassi

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

- **key** (richiesto) — chiave Redis di tipo documento JSON
- **percorso** (obbligatorio) — un percorso JSON
- **index** (obbligatorio) — indice dell'array prima del quale vengono inseriti i valori.
- **json** (obbligatorio) — Valore JSON da aggiungere all'array

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di numeri interi, che rappresenta la nuova lunghezza dell'array in ogni percorso.
- Se un valore è un array vuoto, il valore restituito corrispondente è null.
- Se un valore non è un array, il valore restituito corrispondente è null.
- Errore `OUTOFBOUNDARIES` se l'argomento indice è fuori dai limiti.

Se il percorso è una sintassi limitata:

- Numero intero, la nuova lunghezza dell'array.
- Errore `WRONGTYPE` se il valore nel percorso non è un array.

- Errore `OUTOFBOUNDARIES` se l'argomento indice è fuori dai limiti.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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\"]]"
```

Sintassi limitata del percorso:

```
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

Ottieni la lunghezza dei valori dell'array nel percorso.

Sintassi

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

- **key** (richiesto) — chiave Redis di tipo documento JSON
- **percorso** (opzionale) — un percorso JSON. Se non è fornito, viene ripristinato il percorso predefinito, la radice

Valori restituiti

Se il percorso è una sintassi avanzata:

- Matrice di numeri interi, che rappresenta la lunghezza della matrice in ogni percorso.
- Se un valore non è un array, il valore restituito corrispondente è null.
- Null se la chiave del documento non esiste.

Se il percorso è una sintassi limitata:

- Array di stringhe in blocco. Ogni elemento è un nome chiave nell'oggetto.
- Numero intero, lunghezza dell'array.
- Se sono selezionati più oggetti, il comando restituisce la lunghezza del primo array.
- Errore WRONGTYPE se il valore nel percorso non è un array.
- Errore WRONGTYPE se il percorso non esiste.
- Null se la chiave del documento non esiste.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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)
```

Sintassi limitata del percorso:

```
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

Rimuove e restituisce l'elemento all'indice dall'array. Il prelievo di un array vuoto restituisce null.

Sintassi

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

- **key** (richiesto) — chiave Redis di tipo documento JSON
- **percorso** (opzionale) — un percorso JSON. Se non è fornito, viene ripristinato il percorso predefinito, la radice
- **index** (opzionale) — posizione nell'array da cui iniziare a comparire.
 - Viene ripristinato il valore predefinito -1 se non è fornito, ossia l'ultimo elemento.
 - Un valore negativo indica la posizione dall'ultimo elemento.
 - Gli indici fuori limite vengono arrotondati ai rispettivi limiti dell'array.

Valori restituiti

Se il percorso è una sintassi avanzata:

- Matrice di stringhe di massa, che rappresentano i valori visualizzati in ogni percorso.

- Se un valore è un array vuoto, il valore restituito corrispondente è null.
- Se un valore non è un array, il valore restituito corrispondente è null.

Se il percorso è una sintassi limitata:

- Stringa di massa, che rappresenta il valore JSON visualizzato
- Null se l'array è vuoto.
- Errore WRONGTYPE se il valore nel percorso non è un array.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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\"]]"
```

Sintassi limitata del percorso:

```
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

Taglia gli array in corrispondenza del percorso in modo che diventi sottoarray [inizio, fine], entrambi inclusi.

- Se l'array è vuoto, non eseguire nulla, restituire 0.
- Se start < 0, considerarlo come 0.
- Se end >= size (dimensione dell'array), considerarlo come size-1.
- Se start >= size o start > end, svuotare l'array e restituire 0.

Sintassi

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

- key (richiesto) — chiave Redis di tipo documento JSON
- percorso (obbligatorio) — un percorso JSON
- start (obbligatorio) — indice iniziale, incluso.
- fine (obbligatorio) — indice finale, incluso.

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di numeri interi, che rappresenta la nuova lunghezza dell'array in ogni percorso.
- Se un valore è un array vuoto, il valore restituito corrispondente è null.
- Se un valore non è un array, il valore restituito corrispondente è null.
- Errore `OUTOFBOUNDARIES` se un argomento indice è fuori dai limiti.

Se il percorso è una sintassi limitata:

- Numero intero, la nuova lunghezza dell'array.
- Null se l'array è vuoto.
- Errore `WRONGTYPE` se il valore nel percorso non è un array.
- Errore `OUTOFBOUNDARIES` se un argomento indice è fuori dai limiti.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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\\"]]"
```

Sintassi limitata del percorso:

```
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

Cancellate le matrici o gli oggetti sul percorso.

Sintassi

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

- **key** (richiesto) — chiave Redis di tipo documento JSON
- **percorso** (opzionale) — un percorso JSON. Se non è fornito, viene ripristinato il percorso predefinito, la radice

Valori restituiti

- Numero intero, il numero di container cancellati.

- La cancellazione di un oggetto o un array vuoto rappresenta 0 container cancellati.

Note

La versione Redis 6.2.6.R2 container cancellato.

- La cancellazione di un valore non container restituisce 0.
- Se nel percorso non si trova alcun valore di matrice o oggetto, il comando restituisce 0.

Examples (Esempi)

```
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

Informazioni sul rapporto. Sottocomandi supportati:

- MEMORY <key>[percorso]: segnala l'utilizzo della memoria in byte di un valore JSON. Se non è fornito, viene ripristinato il percorso predefinito, la radice.
- <key>DEPTH [percorso]: riporta la profondità massima del percorso del documento JSON.

Note

Questo sottocomando è disponibile solo utilizzando il motore Redis versione 6.6.6.R2 o successiva.

- FIELDS <key>[percorso]: riporta il numero di campi nel percorso del documento specificato. Se non è fornito, viene ripristinato il percorso predefinito, la radice. Ogni valore JSON non container

viene conteggiato come un singolo campo. Oggetti e array vengono conteggiati ricorsivamente come singolo campo per ognuno dei loro valori JSON contenenti. Ogni valore container, tranne il container radice, viene conteggiato come un campo aggiuntivo.

- HELP: stampa i messaggi di aiuto del comando.

Sintassi

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

Dipende dal sottocomando:

MEMORY

- Se il percorso è una sintassi avanzata:
 - restituisce una matrice di numeri interi, che rappresenta la dimensione della memoria (in byte) del valore JSON in ogni percorso.
 - restituisce un array vuoto se la chiave Redis non esiste.
- Se il percorso è una sintassi limitata:
 - restituisce un numero intero, la dimensione della memoria è il valore JSON in byte.
 - restituisce null se la chiave Redis non esiste.

DEPTH

- Restituisce un numero intero che rappresenta la profondità massima del percorso del documento JSON.
- Restituisce null se la chiave Redis non esiste.

FIELDS

- Se il percorso è una sintassi avanzata:
 - restituisce una matrice di numeri interi, che rappresenta il numero di campi di valore JSON in ogni percorso.
 - restituisce un array vuoto se la chiave Redis non esiste.
- Se il percorso è una sintassi limitata:
 - restituisce un numero intero, numero di campi del valore JSON.

- restituisce null se la chiave Redis non esiste.

AIUTO: restituisce una serie di messaggi di aiuto.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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
```

Sintassi limitata del percorso:

```
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

Elimina i valori JSON nel percorso in una chiave di documento. Se il percorso è la radice, equivale all'eliminazione della chiave da Redis.

Sintassi

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

- **key** (richiesto) — chiave Redis di tipo documento JSON
- **percorso** (opzionale) — un percorso JSON. Se non è fornito, viene ripristinato il percorso predefinito, la radice

Valori restituiti

- Numero di elementi eliminati.
- 0 se la chiave Redis non esiste.
- 0 se il percorso JSON non è valido o non esiste.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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\":[]}"
```

Sintassi limitata del percorso:

```
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 di [JSON.DEL](#)

JSON.GET

Restituisce il JSON serializzato in uno o più percorsi.

Sintassi

```
JSON.GET <key>
[INDENT indentation-string]
[NEWLINE newline-string]
```

```
[SPACE space-string]
[NOESCAPE]
[path ...]
```

- **key (richiesto)** — chiave Redis di tipo documento JSON
- **INDENT/NEWLINE/SPACE** (opzionale): controlla il formato della stringa JSON restituita, ad esempio «pretty print». Il valore predefinito di ognuno è una stringa vuota. Possono essere sovrascritti in qualsiasi combinazione. Possono essere specificati in qualunque ordine.
- **NOESCAPE**: opzionale, può essere presente per la compatibilità con le versioni precedenti e non ha altri effetti.
- **percorso** (opzionale): zero o più percorsi JSON, il valore predefinito è root se non ne viene fornito nessuno. Gli argomenti del percorso devono essere collocati alla fine.

Valori restituiti

Sintassi avanzata del percorso:

Se viene fornito un percorso:

- Restituisce una stringa serializzata di un array di valori.
- Se non è selezionato alcun valore, il comando restituisce un array vuoto.

Se vengono forniti più percorsi:

- Restituisce un oggetto JSON stringato, in cui ogni percorso è una chiave.
- in presenza di sintassi mista e avanzata dei percorsi, il risultato è conforme alla sintassi avanzata.
- Se un percorso non esiste, il valore corrispondente è un array vuoto.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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]"

```

Sintassi limitata del percorso:

```

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

Ottieni JSON serializzati nel percorso da più chiavi di documento. Restituisce null per una chiave o un percorso JSON inesistente.

Sintassi

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

- **key (richiesto)** — Una o più chiavi Redis di tipo documento.

- percorso (obbligatorio) — un percorso JSON

Valori restituiti

- Gamma di corde Bulk. La dimensione dell'array è uguale al numero di chiavi nel comando. Ogni elemento dell'array è popolato con (a) il JSON serializzato situato in base al percorso o (b) Null se la chiave non esiste o il percorso non esiste nel documento o il percorso non è valido (errore di sintassi).
- Se una delle chiavi specificate esiste e non è una chiave JSON, il comando restituisce l'errore WRONGTYPE.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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\ "]"
```

Sintassi limitata del percorso:

```
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

Incrementa i valori numerici sul percorso di un determinato numero.

Sintassi

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

- **key** (richiesto) — chiave Redis di tipo documento JSON
- **percorso** (obbligatorio) — un percorso JSON
- **numero** (richiesto) — un numero

Valori restituiti

Se il percorso è una sintassi avanzata:

- Matrice di stringhe di massa che rappresentano il valore risultante in ogni percorso.
- Se un valore non è un numero, il valore restituito corrispondente è null.
- Errore `WRONGTYPE` se il numero non può essere analizzato.
- Errore `OVERFLOW` se il risultato non rientra nell'intervallo del doppio IEEE a 64 bit.
- `NONEXISTENT` se la chiave del documento non esiste.

Se il percorso è una sintassi limitata:

- Stringa di massa che rappresenta il valore risultante.
- Se sono selezionati più valori array, il comando restituisce il risultato dell'ultimo valore aggiornato.
- Errore `WRONGTYPE` se il valore nel percorso non è un numero.
- Errore `WRONGTYPE` se il numero non può essere analizzato.
- Errore `OVERFLOW` se il risultato non rientra nell'intervallo del doppio IEEE a 64 bit.

- NONEXISTENT se la chiave del documento non esiste.

Examples (Esempi)

Sintassi avanzata del percorso:

```

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}}"

```

Sintassi limitata del percorso:

```

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

Moltiplica i valori numerici sul percorso per un dato numero.

Sintassi

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

- **key** (richiesto) — chiave Redis di tipo documento JSON
- **percorso** (obbligatorio) — un percorso JSON
- **numero** (richiesto) — un numero

Valori restituiti

Se il percorso è una sintassi avanzata:

- Matrice di stringhe di massa che rappresentano il valore risultante in ogni percorso.
- Se un valore non è un numero, il valore restituito corrispondente è null.
- Errore WRONGTYPE se il numero non può essere analizzato.
- Errore OVERFLOW se il risultato non rientra nell'intervallo del doppio IEEE a 64 bit.
- NONEXISTENT se la chiave del documento non esiste.

Se il percorso è una sintassi limitata:

- Stringa di massa che rappresenta il valore risultante.
- Se sono selezionati più valori array, il comando restituisce il risultato dell'ultimo valore aggiornato.
- Errore WRONGTYPE se il valore nel percorso non è un numero.
- Errore WRONGTYPE se il numero non può essere analizzato.
- Errore OVERFLOW se il risultato non rientra nell'intervallo del doppio IEEE a 64 bit.
- NONEXISTENT se la chiave del documento non esiste.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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]"

```

Sintassi limitata del percorso:

```

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

Ottieni il numero di chiavi nei valori dell'oggetto nel percorso.

Sintassi

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

- **key** (richiesto) — chiave Redis di tipo documento JSON
- **percorso** (opzionale) — un percorso JSON. Se non è fornito, viene ripristinato il percorso predefinito, la radice

Valori restituiti

Se il percorso è una sintassi avanzata:

- Matrice di numeri interi, che rappresenta la lunghezza dell'oggetto in ogni percorso.
- Se un valore non è un oggetto, il valore restituito corrispondente è null.
- Null se la chiave del documento non esiste.

Se il percorso è una sintassi limitata:

- Numero intero, numero di chiavi nell'oggetto.
- Se sono selezionati più oggetti, il comando restituisce la lunghezza del primo oggetto.
- Errore `WRONGTYPE` se il valore nel percorso non è un oggetto.
- Errore `WRONGTYPE` se il percorso non esiste.
- Null se la chiave del documento non esiste.

Examples (Esempi)

Sintassi avanzata del percorso:

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

```

Sintassi limitata del percorso:

```

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

Ottieni i nomi delle chiavi nei valori degli oggetti nel percorso.

Sintassi

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

- **key** (richiesto) — chiave Redis di tipo documento JSON
- **percorso** (opzionale) — un percorso JSON. Se non è fornito, viene ripristinato il percorso predefinito, la radice

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di array di stringhe in blocco. Ogni elemento è un array di chiavi in un oggetto corrispondente.
- Se un valore non è un oggetto, il valore restituito corrispondente è un valore vuoto.
- Null se la chiave del documento non esiste.

Se il percorso è una sintassi limitata:

- Array di stringhe in blocco. Ogni elemento è un nome chiave nell'oggetto.
- Se sono selezionati più oggetti, il comando restituisce le chiavi del primo oggetto.
- Errore `WRONGTYPE` se il valore nel percorso non è un oggetto.
- Errore `WRONGTYPE` se il percorso non esiste.
- Null se la chiave del documento non esiste.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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"
```

Sintassi limitata del percorso:

```
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

Restituisce il valore JSON nel percorso indicato in Redis Serialization Protocol (RESP). Se il valore è contenitore, la risposta è array RESP o array annidato.

- Un valore null JSON è mappato alla stringa in blocco null RESP.
- I valori booleani JSON sono mappati alle rispettive stringhe semplici RESP.

- I numeri interi sono mappati a numeri interi RESP.
- I numeri a virgola mobile doppia IEEE a 64 bit sono mappati a stringhe in blocco RESP.
- Le stringhe in blocco sono mappate a stringhe in blocco.
- Gli array JSON sono rappresentati come array RESP, dove il primo elemento è la semplice stringa `[`, seguita dagli elementi dell'array.
- Gli oggetti JSON sono rappresentati come array RESP, dove il primo elemento è la semplice stringa `{`, seguita da coppie chiave-valore, ognuna delle quali è una stringa di massa RESP.

Sintassi

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

- `key` (richiesto) — chiave Redis di tipo documento JSON
- `percorso` (opzionale) — un percorso JSON. Se non è fornito, viene ripristinato il percorso predefinito, la radice

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di array. Ogni elemento dell'array rappresenta la forma RESP del valore in un unico percorso.
- Array vuoto se la chiave del documento non esiste.

Se il percorso è una sintassi limitata:

- Array, che rappresenta la forma in blocco del valore nel percorso.
- Null se la chiave del documento non esiste.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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"

Sintassi limitata del percorso:

```
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

Imposta i valori JSON nel percorso.

Se il percorso richiede un membro oggetto:

- Se l'elemento padre non esiste, il comando restituirà un errore NONEXISTENT.
- Se l'elemento padre esiste ma non è un oggetto, il comando restituirà ERROR.
- Se l'elemento padre esiste ed è un oggetto:
 - Se il membro non esiste, un nuovo membro verrà accodato all'oggetto padre se e solo se l'oggetto padre è l'ultimo figlio nel percorso. Altrimenti, il comando restituirà un errore NONEXISTENT.
 - Se il membro esiste, il suo valore verrà sostituito dal valore JSON.

Se il percorso richiede un indice di array:

- Se l'elemento padre non esiste, il comando restituirà un errore INESISTENTE.
- Se l'elemento padre esiste ma non è un array, il comando restituirà ERROR.

- Se l'elemento padre esiste ma l'indice è fuori dai limiti, il comando restituirà l'errore `OUTOFBOUNDERS`.
- Se l'elemento padre esiste e l'indice è valido, l'elemento verrà sostituito dal nuovo valore JSON.

Se il percorso richiede un oggetto o un array, il valore (oggetto o array) verrà sostituito dal nuovo valore JSON.

Sintassi

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

[NX | XX] Dove puoi avere 0 o 1 identificatori [NX | XX]

- `key` (richiesto) — chiave Redis di tipo documento JSON
- `percorso` (obbligatorio) — percorso JSON. Per una nuova chiave Redis, il percorso JSON deve essere la radice `."`.
- `NX` (facoltativo) — Se il percorso è la radice, impostare il valore solo se la chiave Redis non esiste, ossia se il documento nuovo. Se il percorso non è la radice, imposta il valore solo se il percorso non esiste, ad esempio inserisci un valore nel documento.
- `XX` (facoltativo) — Se il percorso è la radice, impostare il valore solo se la chiave Redis esiste, cioè sostituisci il documento esistente. Se il percorso non è la radice, imposta il valore solo se il percorso esiste, ad esempio aggiorna il valore esistente.

Valori restituiti

- Stringa semplice `'OK'` se l'esito è positivo.
- Null se la condizione `NX` o `XX` non viene soddisfatta.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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]}"
```

Sintassi limitata del percorso:

```
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

Aggiungi una stringa alle stringhe JSON nel percorso.

Sintassi

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

- **key** (richiesto) — chiave Redis di tipo documento JSON
- **percorso** (opzionale) — un percorso JSON. Se non è fornito, viene ripristinato il percorso predefinito, la radice
- **json_string** (obbligatorio) — Rappresentazione JSON di una stringa. Nota che una stringa JSON deve essere citata, ad esempio `"foo"`.

Valori restituiti

Se il percorso è una sintassi avanzata:

- Matrice di numeri interi, che rappresenta la nuova lunghezza della stringa in ogni percorso.
- Se un valore nel percorso non è una stringa, il valore restituito corrispondente è null.
- SYNTAXERR errore se l'argomento json di input non è una stringa JSON valida.
- NONEXISTENT errore se il percorso non esiste.

Se il percorso è una sintassi limitata:

- Numero intero, la nuova lunghezza della stringa.
- Se sono selezionati più valori array, il comando restituisce la nuova lunghezza dell'ultima stringa aggiornata.
- Errore WRONGTYPE se il valore nel percorso non è una stringa.
- Errore WRONGTYPE se l'argomento json di input non è una stringa JSON valida.
- Errore NONEXISTENT se il percorso non esiste.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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)
```

Sintassi limitata del percorso:

```
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

Ottieni le lunghezze dei valori della stringa JSON nel percorso.

Sintassi

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

- **key** (richiesto) — chiave Redis di tipo documento JSON
- **percorso** (opzionale) — un percorso JSON. Se non è fornito, viene ripristinato il percorso predefinito, la radice

Valori restituiti

Se il percorso è una sintassi avanzata:

- **Matrice di numeri interi**, che rappresenta la lunghezza del valore della stringa in ogni percorso.
- Se un valore non è una stringa, il valore restituito corrispondente è null.

- Null se la chiave del documento non esiste.

Se il percorso è una sintassi limitata:

- Numero intero, la lunghezza della stringa.
- Se sono selezionati più valori stringa, il comando restituisce la lunghezza della prima stringa.
- Errore WRONGTYPE se il valore nel percorso non è una stringa.
- Errore NONEXISTENT se il percorso non esiste.
- Null se la chiave del documento non esiste.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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)
```

Sintassi limitata del percorso:

```
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

Cambia i valori booleani tra vero e falso nel percorso.

Sintassi

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

- **key** (richiesto) — chiave Redis di tipo documento JSON
- **percorso** (opzionale) — un percorso JSON. Se non è fornito, viene ripristinato il percorso predefinito, la radice

Valori restituiti

Se il percorso è una sintassi avanzata:

- Matrice di numeri interi (0 - falso, 1 - vero) che rappresentano il valore booleano risultante in ogni percorso.
- Se un valore non è un valore non è un valore restituito è null.
- **NONEXISTENT** se la chiave del documento non esiste.

Se il percorso è una sintassi limitata:

- Stringa («true» /"false») che rappresenta il valore booleano risultante.
- **NONEXISTENT** se la chiave del documento non esiste.
- **WRONGTYPE** errore se il valore nel percorso non è un valore booleano.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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)
```

Sintassi limitata del percorso:

```
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

Tipo di rapporto dei valori nel percorso indicato.

Sintassi

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

- **key** (richiesto) — chiave Redis di tipo documento JSON
- **percorso** (opzionale) — un percorso JSON. Se non è fornito, viene ripristinato il percorso predefinito, la radice

Valori restituiti

Se il percorso è una sintassi avanzata:

- Matrice di stringhe, che rappresenta il tipo di valore in ogni percorso. Il tipo è uno di {"null", "boolean", "string", "number", "integer", "object" e "array"}.
- Se un percorso non esiste, il valore restituito corrispondente è null.
- Array vuoto se la chiave del documento non esiste.

Se il percorso è una sintassi limitata:

- Stringa, tipo di valore
- Null se la chiave del documento non esiste.
- Null se il percorso JSON non è valido o non esiste.

Examples (Esempi)

Sintassi avanzata del percorso:

```
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
```

Sintassi limitata del percorso:

```
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
```

Tagging delle risorse MemoryDB

Per semplificare la gestione delle risorse MemoryDB, è possibile assegnare metadati personalizzati a ciascuna risorsa sotto forma di tag. I tag consentono di categorizzare le tue risorse AWS in modi diversi, ad esempio, per scopo, proprietario o ambiente. Questa caratteristica è molto utile quando hai tante risorse dello stesso tipo in quanto puoi rapidamente individuare una risorsa specifica in base ai tag assegnati. Questo argomento descrive i tag e mostra come crearli.

Warning

Come da best practice, è consigliabile non includere dati sensibili nei tag.

Nozioni di base sui tag

Un tag è un'etichetta che assegni a una risorsa AWS. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili. I tag consentono di categorizzare le tue risorse AWS in modi diversi, ad esempio, per scopo o proprietario. Ad esempio, puoi definire un set di tag per i cluster del tuo account MemoryDB e monitorare così ogni proprietario del cluster e il gruppo di utenti.

Ti consigliamo di creare un set di chiavi di tag in grado di soddisfare i requisiti di ciascun tipo di risorsa. Tramite un set di chiavi di tag coerente la gestione delle risorse risulta notevolmente semplificata. Puoi cercare e filtrare le risorse in base ai tag aggiunti. Per ulteriori informazioni su come implementare una strategia efficace di applicazione di tag alle risorse, consulta [ilAWSwhitepaper Tagging Best Practice](#).

I tag non hanno alcun significato semantico per MemoryDB e vengono interpretati rigorosamente come una stringa di caratteri. Inoltre, i tag non vengono assegnati automaticamente alle risorse. Puoi modificare chiavi e valori di tag e rimuovere tag da una risorsa in qualsiasi momento. È possibile impostare il valore di un tag su null. Se aggiungi un tag con la stessa chiave di un tag esistente a una risorsa specifica, il nuovo valore sovrascrive quello precedente. Se elimini una risorsa, verranno eliminati anche tutti i tag associati alla risorsa.

Puoi lavorare con i tag utilizzando l'AWS Management Console, ilAWS CLIe l'API MemoryDB.

Se utilizzi (IAM), puoi controllare quali utenti nel tuo account AWS dispongono dell'autorizzazione per creare, modificare o eliminare i tag. Per ulteriori informazioni, consulta la pagina [Autorizzazioni a livello di risorsa](#).

Risorse cui è possibile associare tag

Puoi assegnare tag alla maggior parte delle risorse MemoryDB già esistenti nel tuo account. Nella tabella seguente sono elencate le risorse che supportano il tagging. Se usi il comando AWS Management Console, puoi applicare tag a risorse utilizzando l'[Editor di tag](#). Alcune schermate relative alle risorse ti permettono di specificare i tag per una risorsa quando crei la risorsa, ad esempio un tag con la chiave con nome e un valore specificato. Nella maggior parte dei casi, la console applica i tag subito dopo la creazione della risorsa, anziché durante il processo di creazione. La console può organizzare le risorse in base alNome di questo tag non ha un significato semantico per il servizio MemoryDB.

Inoltre, alcune operazioni per la creazione di risorse ti consentono di specificare tag per una risorsa durante la sua creazione. Se i tag non possono essere applicati durante la creazione della risorsa, eseguiamo il rollback del processo di creazione della risorsa. Ciò fa sì che le risorse vengano create

con i tag oppure che non vengano create affatto, nonché che nessuna risorsa sia mai sprovvista di tag. Il tagging delle risorse in fase di creazione ti permette di evitare di eseguire script di tagging personalizzati dopo la creazione delle risorse.

Se utilizzi l'API Amazon MemoryDB, laAWSCLI o unAWSSDK, è possibile utilizzare l'Tagsparametro sull'azione API MemoryDB pertinente per applicare i tag. Questi sono:

- CreateCluster
- CopySnapshot
- CreateParameterGroup
- CreateSubnetGroup
- CreateSnapshot
- CreateACL
- CreateUser

Nella seguente tabella sono descritte le risorse MemoryDB a cui puoi associare i tag, nonché le risorse che possono essere associate a tag utilizzando l'API MemoryDB, ilAWSCLI o unAWSSDK.

Supporto del tagging per le risorse MemoryDB

support dei tag	Supporta l'applicazione di tag in fase di creazione
Sì	Sì
Sì	Sì
Sì	Sì
Sì	Sì
Sì	Sì
Sì	Sì

Puoi applicare autorizzazioni basate su tag a livello di risorsa nelle policy IAM alle operazioni dell'API MemoryDB che supportano il tagging in fase di creazione per implementare un controllo granulare sugli utenti e sui gruppi che associano tag alle risorse in fase di creazione. Le risorse vengono adeguatamente protette dalla creazione, ovvero tag che vengono applicati immediatamente alle risorse. Pertanto qualsiasi autorizzazione basata su tag a livello di risorsa che controlla l'uso delle risorse risulta immediatamente valida. Le risorse possono essere monitorate e segnalate con maggiore precisione. Puoi applicare l'uso del tagging alle nuove risorse e controllare quali chiavi e valori di tag sono impostati per le risorse.

Per ulteriori informazioni, consulta la pagina [Esempio: assegnazione di tag alle risorse](#) .

Per ulteriori informazioni sul tagging delle risorse per la fatturazione, vedere [Monitoraggio dei costi con i tag di allocazione dei costi](#).

Tagging di cluster e istantanee

Le seguenti regole si applicano alle etichette come parte delle operazioni di richiesta:

- **CreateCluster** :
 - Se il file `--cluster-name` viene fornito:

Se i tag sono inclusi nella richiesta, il cluster verrà contrassegnato con tag.
 - Se il file `--snapshot-name` viene fornito:

Se i tag sono inclusi nella richiesta, il cluster verrà contrassegnato solo con tali tag. Se nella richiesta non sono inclusi tag, i tag snapshot verranno aggiunti al cluster.
- **CreateSnapshot** :
 - Se il file `--cluster-name` viene fornito:

Se i tag sono inclusi nella richiesta, solo i tag di richiesta verranno aggiunti allo snapshot. Se nella richiesta non sono inclusi tag, i tag cluster verranno aggiunti allo snapshot.
 - **Snapshot automatiche**

I tag verranno propagati dai tag cluster.
- **CopySnapshot** :

Se i tag sono inclusi nella richiesta, solo i tag di richiesta verranno aggiunti allo snapshot. Se nella richiesta non sono inclusi tag, i tag snapshot di fonte verranno aggiunti allo snapshot copiato.
- **TagResourceUntagResource**:

I tag verranno aggiunti/rimossi dalla risorsa.

Limitazioni applicate ai tag

Si applicano le seguenti limitazioni di base ai tag:

- Numero massimo di tag per risorsa: 50
- Per ciascuna risorsa, ogni chiave del tag deve essere univoca e ogni chiave del tag può avere un solo valore.
- La lunghezza massima della chiave è 128 caratteri Unicode in formato UTF-8.
- La lunghezza massima del valore è 256 caratteri Unicode in formato UTF-8.
- Sebbene MemoryDB consenta l'uso di qualsiasi carattere nei tag, altri servizi possono essere restrittivi. I caratteri consentiti nei servizi sono: lettere, numeri e spazi rappresentabili in formato UTF-8 e i seguenti caratteri speciali + - = . _ : / @.
- Per chiavi e valori di tag viene fatta la distinzione tra maiuscole e minuscole.
- Il prefisso `aws :` è riservato per l'uso di AWS. Se il tag ha una chiave di tag con questo prefisso, non puoi modificare o eliminare la chiave o il valore de tag. I tag con il prefisso `aws :` non vengono conteggiati per il limite del numero di tag per risorsa.

Non puoi interrompere, arrestare o eliminare una risorsa solo sulla base dei relativi tag. Devi specificare il relativo identificatore. Ad esempio, per eliminare gli snapshot associato a una chiave di tag denominata `DeleteMe`, devi utilizzare l'operazione `DeleteSnapshot` con gli identificatori di risorsa degli snapshot, ad esempio `snap-1234567890abcdef0`.

Per ulteriori informazioni sulle risorse MemoryDB che è possibile assegnare tag, vedere [Risorse cui è possibile associare tag](#).

Esempio: assegnazione di tag alle risorse

- Aggiunta di tag a un cluster.

```
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"
```

- Creazione di un cluster tramite i tag.

```
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"
```

- Esempio: creazione di uno snapshot con tag

In questo caso, se si aggiungono tag su richiesta, anche se il cluster contiene tag, lo snapshot riceverà solo i tag di richiesta.

```
aws memorydb create-snapshot \  
--cluster-name testing-tags \  
--snapshot-name bkp-testing-tags-mycluster \  
--tags Key="work",Value="foo"
```

Monitoraggio dei costi con i tag di allocazione dei costi

Quando aggiungi i tag di allocazione dei costi alle tue risorse in MemoryDB per Redis, puoi tenere traccia dei costi raggruppando le spese nelle fatture mediante i valori dei tag della risorsa.

Un tag di allocazione dei costi MemoryDB è una coppia chiave-valore definita e associata a una risorsa MemoryDB. La chiave e il valore fanno distinzione tra maiuscole e minuscole. Puoi utilizzare una chiave di tag per definire una categoria e il valore come una voce di tale categoria. Ad esempio, puoi definire una chiave di tag `CostCenter` e un valore di tag `10010`, a indicare che la risorsa è assegnata al centro di costo 10010. È anche possibile usare i tag per indicare le risorse come risorse utilizzate a scopo di test o produzione tramite una chiave, ad esempio `Environment`, e tramite valori, ad esempio `test` o `production`. È consigliabile utilizzare un set coerente di chiavi di tag per agevolare il monitoraggio dei costi associati alle risorse.

Per organizzare le fatture AWS al fine di riflettere la struttura dei costi, puoi utilizzare i tag di allocazione dei costi. Per eseguire questa operazione, registrati per far sì che la fattura del tuo account AWS includa i valori di chiave di tag. Per visualizzare il costo delle risorse combinate, puoi organizzare le informazioni di fatturazione in base alle risorse con gli stessi valori di chiave di tag. Puoi ad esempio applicare tag a numerose risorse con un nome di applicazione specifico, quindi

organizzare le informazioni di fatturazione per visualizzare il costo totale dell'applicazione in più servizi.

Puoi anche combinare i tag per monitorare i costi con un livello di dettagli maggiore. Ad esempio, per monitorare i costi di servizio per regione, puoi utilizzare le chiavi di tag `Service` e `Region`. Su una risorsa potresti avere i valori `MemoryDB` e `Asia Pacific (Singapore)`, mentre su un'altra risorsa potresti avere i valori `MemoryDB` e `Europe (Frankfurt)`. Puoi quindi vedere i tuoi costi totali `MemoryDB` suddivisi tra le regioni. Per ulteriori informazioni, consulta la pagina sull'[utilizzo dei tag per l'allocazione dei costi](#) nella Guida per l'utente di AWS Billing.

Puoi aggiungere i tag di allocazione dei costi `MemoryDB` ai cluster `MemoryDB`. Quando aggiungi, elenchi, modifichi, copi o rimuovi un tag, l'operazione viene applicata solo al cluster specificato.

Caratteristiche dei tag di allocazione dei costi `MemoryDB`

- I tag di allocazione dei costi vengono applicati alle risorse `MemoryDB` specificate nelle operazioni API e CLI come ARN. Il tipo di risorsa sarà un cluster.

Formato ARN: `arn:aws:memorydb:<region>:<customer-id>:<resource-type>/<resource-name>`

ARN di esempio: `arn:aws:memorydb:us-east-1:1234567890:cluster/my-cluster`

- La chiave di tag corrisponde al nome obbligatorio del tag. Il valore della stringa della chiave può essere composto da 1 a 128 caratteri Unicode e non può avere il prefisso `aws:`. La stringa può contenere solo il set di lettere, numeri, spazi vuoti, caratteri di sottolineatura (`_`), punti (`.`), virgole (`:`), barre rovesciate (`\`), segni di uguale (`=`), più (`+`), trattini (`-`) o chioccioline (`@`).
- Un valore tag è il valore opzionale del tag. Il valore di stringa del valore può essere composto da 1 a 256 caratteri Unicode e non può avere il prefisso `aws:`. La stringa può contenere solo il set di lettere, numeri, spazi vuoti, caratteri di sottolineatura (`_`), punti (`.`), virgole (`:`), barre rovesciate (`\`), segni di uguale (`=`), più (`+`), trattini (`-`) o chioccioline (`@`).
- Una risorsa `MemoryDB` può avere un massimo di 50 tag.
- I valori non devono essere necessariamente univoci in un set di tag. Ad esempio, puoi avere un set di tag dove le chiavi `Service` e `Application` hanno entrambe il valore `MemoryDB`.

AWS non applica significati semantici ai tag. I tag sono interpretati prettamente come stringhe di caratteri. AWS non imposta automaticamente alcun tag su nessuna risorsa `MemoryDB`.

Gestione dei tag di allocazione dei costi tramite la console AWS CLI

Puoi utilizzare l'AWS CLI per aggiungere, modificare o rimuovere i tag di allocazione dei costi.

Arn di esempio: `arn:aws:memorydb:us-east-1:1234567890:cluster/my-cluster`

Argomenti

- [Elenco dei tag utilizzando la AWS CLI](#)
- [Aggiunta di tag utilizzando la AWS CLI](#)
- [Modifica dei tag utilizzando l' AWS CLI](#)
- [Rimozione dei tag utilizzando l' AWS CLI](#)

Elenco dei tag utilizzando la AWS CLI

Puoi utilizzare il plugin AWS CLI per elencare i tag su una risorsa MemoryDB esistente tramite l'[list-tags](#) operazione.

Il seguente codice utilizza il AWS CLI per elencare i tag sul cluster MemoryDB `my-cluster` nella regione `us-east-1`.

Per Linux, macOS o Unix:

```
aws memorydb list-tags \  
  --resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster
```

Per Windows:

```
aws memorydb list-tags ^  
  --resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster
```

L'output di questa operazione sarà simile al seguente, una lista di tutti i tag sulla risorsa.

```
{  
  "TagList": [  
    {  
      "Value": "10110",  
      "Key": "CostCenter"  
    },  
    {
```

```

    "Value": "EC2",
    "Key": "Service"
  }
]
}

```

Se non sono presenti tag sulla risorsa, l'output sarà vuotoTagList.

```

{
  "TagList": []
}

```

Per ulteriori informazioni, consulta la `.AWS CLI` per MemoryDB [list-tags](#).

Aggiunta di tag utilizzando la AWS CLI

Puoi utilizzare il plugin `AWS CLI` per aggiungere i tag a una risorsa MemoryDB esistente tramite l'[tag-resource](#) Funzionamento CLI. Se la nuova chiave di tag non esiste sulla risorsa, la chiave e il valore vengono aggiunti alla risorsa. Se la chiave esiste già sulla risorsa, il valore associato a quella chiave viene aggiornato al nuovo valore.

Il seguente codice utilizza il `AWS CLI` per aggiungere le chiavi `Service` e `Region` con i valori `memorydb` e `us-east-1` rispettivamente al cluster `my-cluster` nella regione `us-east-1`.

Per 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

```

Per 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

```

L'output di questa operazione sarà simile al seguente, una lista di tutti i tag sulla risorsa in base all'operazione.

```
{
  "TagList": [
    {
      "Value": "memorydb",
      "Key": "Service"
    },
    {
      "Value": "us-east-1",
      "Key": "Region"
    }
  ]
}
```

Per ulteriori informazioni, consulta la `.AWS CLI` per MemoryDB [tag-resource](#).

Puoi anche utilizzare l'`AWS CLI` per aggiungere i tag a un cluster quando crei un nuovo cluster tramite l'operazione [create-cluster](#).

Modifica dei tag utilizzando l' AWS CLI

Puoi utilizzare il plugin `AWS CLI` per modificare i tag su un cluster MemoryDB.

Per modificare i tag:

- Utilizza [tag-resource](#) per aggiungere un nuovo tag e valore o per modificare il valore associato a un tag esistente.
- Utilizza [untag-resource](#) per rimuovere i tag specificati dalla risorsa.

L'output da entrambe le operazioni sarà un elenco di tag e i relativi valori sul cluster specificato.

Rimozione dei tag utilizzando l' AWS CLI

Puoi utilizzare il plugin `AWS CLI` per rimuovere i tag da un cluster MemoryDB tramite l'[untag-resource](#) operazione.

Il seguente codice utilizza il `AWS CLI` per rimuovere i tag con i tasti `Service` e `Region` dal cluster `my-cluster` nella regione Stati Uniti orientali 1.

Per Linux, macOS o Unix:

```
aws memorydb untag-resource \
```

```
--resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster \  
--tag-keys Region Service
```

Per Windows:

```
aws memorydb untag-resource ^  
--resource-arn arn:aws:memorydb:us-east-1:0123456789:cluster/my-cluster ^  
--tag-keys Region Service
```

L'output di questa operazione sarà simile al seguente, una lista di tutti i tag sulla risorsa in base all'operazione.

```
{  
  "TagList": []  
}
```

Per ulteriori informazioni, consulta la [.AWS CLI per MemoryDB untag-resource](#).

Gestione dei tag di allocazione dei costi tramite l'API MemoryDB

Puoi utilizzare l'API MemoryDB per aggiungere, modificare o rimuovere i tag di allocazione dei costi.

I tag di allocazione dei costi vengono applicati a MemoryDB per i cluster. Il cluster a cui aggiungere tag viene specificato mediante un Amazon Resource Name (ARN).

Arn di esempio: `arn:aws:memorydb:us-east-1:1234567890:cluster/my-cluster`

Argomenti

- [Elenco dei tag utilizzando l'API MemoryDB](#)
- [Aggiunta di tag utilizzando l'API MemoryDB](#)
- [Modifica dei tag utilizzando l'API MemoryDB](#)
- [Rimozione dei tag utilizzando l'API MemoryDB](#)

Elenco dei tag utilizzando l'API MemoryDB

È possibile utilizzare l'API MemoryDB per elencare i tag su una risorsa esistente tramite la [ListTags](#) operazione.

Il codice seguente utilizza l'API MemoryDB per elencare i tag sulla risorsa `my-cluster` nella regione Stati Uniti orientali 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>
```

Aggiunta di tag utilizzando l'API MemoryDB

È possibile utilizzare l'API MemoryDB per aggiungere i tag a un cluster MemoryDB esistente tramite l'API MemoryDB tramite la [TagResource](#) operazione. Se la nuova chiave di tag non esiste sulla risorsa, la chiave e il valore vengono aggiunti alla risorsa. Se la chiave esiste già sulla risorsa, il valore associato a quella chiave viene aggiornato al nuovo valore.

Il codice seguente utilizza l'API MemoryDB per aggiungere le chiavi `Service` e `Region` con i valori `memorydb` e `us-east-1` rispettivamente alla risorsa `my-cluster` nella regione Stati Uniti orientali 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>
```

Per ulteriori informazioni, consulta [TagResource](#).

Modifica dei tag utilizzando l'API MemoryDB

È possibile utilizzare l'API MemoryDB per modificare i tag su un cluster MemoryDB.

Per modificare il valore di un tag:

- Utilizzare l'operazione [TagResource](#) per aggiungere un nuovo tag e valore o per modificare il valore associato a un tag esistente.
- Utilizzare [UntagResource](#) per rimuovere i tag dalla risorsa.

L'output da entrambe le operazioni sarà un elenco di tag e dei relativi valori sulla risorsa specificata.

Rimozione dei tag utilizzando l'API MemoryDB

È possibile utilizzare l'API MemoryDB per rimuovere i tag da un cluster MemoryDB esistente tramite la [UntagResource](#) operazione.

Il codice seguente utilizza l'API MemoryDB per rimuovere i tag con le chiavi `Service` e `Region` dal cluster `my-cluster` nella regione `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>
```

Gestione della manutenzione

Ogni cluster ha una finestra di manutenzione settimanale durante la quale vengono applicate le modifiche al sistema. Se non specifichi una finestra di manutenzione preferita quando crei o modifichi un cluster, MemoryDB ne assegna una di 60 minuti all'interno dell'intervallo di manutenzione della tua regione, in un giorno della settimana selezionato casualmente.

La finestra di manutenzione di 60 minuti viene selezionata a caso da un blocco di tempo di 8 ore per regione. La seguente tabella elenca i blocchi temporali per ciascuna regione da cui sono assegnate le finestre di manutenzione predefinite. È possibile scegliere una finestra di manutenzione personalizzata, anche non compresa nel blocco della regione.

Codice regione	Nome della Regione	Finestra di manutenzione della regione
ap-northeast-1	Asia Pacific (Tokyo) Region	13:00 - 21:00 UTC
ap-northeast-2	Regione Asia Pacifico (Seoul)	12:00 - 20:00 UTC
ap-south-1	Asia Pacific (Mumbai) Region	17:30-1:30 UTC
ap-southeast-1	Asia Pacific (Singapore) Region	14:00 - 22:00 UTC
ap-east-1	Regione Asia Pacifico (Hong Kong)	13:00 - 21:00 UTC
ap-southeast-2	Asia Pacific (Sydney) Region	12:00 - 20:00 UTC
cn-north-1	Regione Cina (Pechino)	14:00 - 22:00 UTC
cn-northwest-1	Regione Cina (Ningxia)	14:00 - 22:00 UTC
eu-west-3	Regione UE (Parigi)	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	Regione Canada (Centrale)	03:00 - 11:00 UTC
us-east-1	US East (N. Virginia) Region	03:00 - 11:00 UTC
us-east-1	US East (Ohio) Region	04:00–12:00 UTC
us-west-1	US West (N. California) Region	06:00 - 14:00 UTC
us-west-2	US West (Oregon) Region	06:00 - 14:00 UTC

Modifica della finestra di manutenzione di un cluster

La finestra di manutenzione deve avvenire nel momento dell'utilizzo più basso e pertanto potrebbe essere necessario apportare modifiche di tanto in tanto. Puoi modificare il cluster e specificare un intervallo di tempo di 24 ore al massimo durante il quale si verifichino le attività di manutenzione richieste. Qualsiasi modifica del cluster richiesta, ma posticipata o in sospeso, viene apportata durante questo lasso di tempo.

Ulteriori informazioni

Per informazioni sulle finestre di manutenzione e la sostituzione dei nodi, consulta a seguire:

- [Sostituzione dei nodi](#):-Gestione della sostituzione dei nodi
- [Modifica di un cluster MemoryDB](#)— Modifica della finestra di manutenzione di un cluster

Best practice

Di seguito sono indicate le best practice consigliate per l'utilizzo di MemoryDB. Se seguite, tali best practice consentono di migliorare prestazioni e affidabilità del cluster.

Argomenti

- [Comandi Redis limitati](#)
- [Resilienza in MemoryDB per Redis](#)
- [Best practice: Pub/Sub e multiplexdi I/O avanzato](#)
- [Best practice. Dimensionamento di cluster online](#)

Comandi Redis limitati

Per offrire un'esperienza di servizio gestito, MemoryDB limita l'accesso a comandi che richiedono privilegi avanzati. I seguenti comandi non sono disponibili:

- `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`

Resilienza in MemoryDB per Redis

L'infrastruttura globale di AWS è basata su regioni e zone di disponibilità AWS. AWS Le Regioni forniscono più zone di disponibilità fisicamente separate e isolate che sono connesse tramite reti altamente ridondanti, a bassa latenza e velocità effettiva elevata. Con le zone di disponibilità, è possibile progettare e gestire le applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo.

Per ulteriori informazioni sulle regioni AWS e sulle zone di disponibilità, consulta [Infrastruttura globale di AWS](#).

Oltre a AWS MemoryDB per Redis offre numerose funzionalità per supportare la resilienza dei dati e le esigenze di snapshot.

Argomenti

- [Limitazione dell'impatto degli errori](#)

Limitazione dell'impatto degli errori

Quando si pianifica l'implementazione di MemoryDB per Redis, è necessario pianificare in modo tale che gli errori abbiano un impatto minimo sull'applicazione e sui dati. Questa sezione, organizzata in più argomenti, illustra cosa fare per proteggere l'applicazione e i dati in caso di errori.

Limitazione degli errori: Cluster MemoryDB

Un cluster MemoryDB è composto da un solo nodo primario, dal quale e sul quale l'applicazione può leggere e scrivere, e da massimo cinque nodi di replica di sola lettura. Tuttavia, si consiglia vivamente di utilizzare almeno 1 replica per un'elevata disponibilità. Ogni volta che i dati sono scritti sul nodo primario, sono persistenti nel registro delle transazioni e aggiornati in modo asincrono sui nodi di replica.

Quando una replica di lettura genera un errore

1. MemoryDB rileva la replica non riuscita.
2. MemoryDB porta offline il nodo non riuscito.
3. MemoryDB avvia e fornisce un nodo sostitutivo nella stessa zona di disponibilità.
4. Il nuovo nodo si sincronizza con il registro delle transazioni.

Nel frattempo, l'applicazione può continuare a leggere e scrivere avvalendosi degli altri nodi.

MemoriaDB Multi-AZ

Se Multi-AZ è attivato sui cluster MemoryDB, un nodo primario non riuscito sarà rilevato e sostituito automaticamente,

1. MemoryDB rileva l'errore del nodo primario.
2. MemoryDB viene eseguito il failover su una replica dopo aver assicurato che sia coerente con il primario non riuscito.
3. MemoryDB rende attiva una replica nella zona di disponibilità del nodo primario non riuscito.
4. Il nuovo nodo si sincronizza con il registro delle transazioni.

Il failover su un nodo di replica è un processo generalmente più veloce della creazione con il provisioning di un nuovo nodo primario. Ciò significa che l'applicazione può riprendere a scrivere sul nodo primario più rapidamente.

Per ulteriori informazioni, consultare [Riduzione al minimo dei tempi di inattività in MemoryDB con Multi-AZ](#).

Best practice: Pub/Sub e multiplexdi I/O avanzato

Quando si utilizza Redis versione 7 o successive, si consiglia di utilizzare [Pub/Sub partizionato](#). È inoltre possibile migliorare la velocità di trasmissione effettiva e la latenza utilizzando il [multiplexing I/O avanzato](#), che è automaticamente disponibile quando si utilizza Redis versione 7 o successive e che non richiede modifiche al client. È ideale per i carichi di lavoro pub/sub, che spesso sono legati alla velocità di trasmissione effettiva con più connessioni client.

Best practice. Dimensionamento di cluster online

Il resharding implica l'aggiunta e la rimozione di partizioni o nodi nel cluster e la redistribuzione di spazi chiave. Diversi fattori hanno pertanto impatto sull'operazione di resharding, come il carico sul cluster, l'utilizzo della memoria e la dimensione complessiva dei dati. Per un'esperienza ottimale, ti consigliamo di attenerci a tutte le best practice relative al cluster per una distribuzione uniforme dei modelli di carico di lavoro. È inoltre consigliabile completare i passaggi indicati di seguito.

Prima di avviare il resharding, ti consigliamo di effettuare quanto segue:

- Testa la tua applicazione - Testa il comportamento della tua applicazione durante il resharding in un ambiente di gestione temporanea, se possibile.
- Ottieni una notifica immediata dei problemi di dimensionamento - Il resharding è un'operazione che richiede notevoli risorse di calcolo. Per questo motivo, durante il resharding consigliamo di mantenere l'utilizzo della CPU inferiore all'80% sulle istanze multicore e inferiore al 50% sulle istanze single core. Esegui il monitoraggio dei parametri di MemoryDB e avvia il resharding prima che l'applicazione inizi a riscontrare problemi di dimensionamento. Parametri utili da considerare sono `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections`, `NewConnections`, `FreeableMemory`, `SwapUsage` e `BytesUsedForMemoryDB`.
- Verifica che sia disponibile memoria sufficiente per il dimensionamento - Se esegui il dimensionamento, assicurati che la memoria libera disponibile sulle partizioni da conservare sia almeno 1,5 volte quella utilizzata sulle partizioni che desideri rimuovere.
- Avvia il resharding durante orari non di punta Ciò consente di ridurre la latenza e l'impatto sulla velocità effettiva per il client durante l'operazione di resharding. In questo modo, il resharding viene inoltre completato più rapidamente, in quanto è possibile utilizzare più risorse per la redistribuzione degli slot.
- Analizza il comportamento di timeout del client - Alcuni client potrebbero presentare una latenza più elevata durante il dimensionamento del cluster online. Può essere utile configurare la libreria client con un timeout maggiore, in quanto aumenta il tempo a disposizione del sistema per eseguire

la connessione, anche in caso di condizioni di carico più elevato sul server. In alcuni casi è possibile che si desideri aprire un numero elevato di connessioni al server. In questi casi considera la necessità di aggiungere backoff esponenziale alla logica di riconnessione. In questo modo è possibile evitare l'aumento di nuove connessioni eseguite contemporaneamente sul server.

Durante il resharding, ti consigliamo di effettuare quanto segue:

- Evita comandi che richiedono un elevato utilizzo delle risorse - Evita di eseguire operazioni di I/O e calcolo intensive, come i comandi KEYS e SMEMBERS. Suggeriamo l'utilizzo di questo approccio perché queste operazioni aumentano il carico sul cluster e hanno impatto sulle prestazioni del cluster. Utilizza i comandi SCAN e SSCAN.
- Segui le best practice Lua - Evita script Lua di lunga durata e dichiara sempre in anticipo le chiavi utilizzate degli script Lua. Consigliamo questo approccio per determinare che lo script Lua non utilizza comandi tra slot. Assicurati che le chiavi utilizzate negli script Lua appartengano allo stesso slot.

Dopo il resharding, tieni presente quanto segue:

- Il dimensionamento potrebbe riuscire parzialmente se la memoria disponibile nelle partizioni di destinazione non è sufficiente. In tal caso, controlla la memoria disponibile e prova di nuovo a eseguire l'operazione, se necessario.
- Per gli slot con elementi di grandi dimensioni non viene eseguita la migrazione. In particolare, la migrazione non viene eseguita per gli slot con elementi di dimensioni maggiori di 256 MB dopo la serializzazione.
- FLUSHALL e FLUSHDB i comandi non sono supportati all'interno degli script Lua durante un'operazione di resharding.

Informazioni sulla replica MemoryDB

MemoryDB implementa la replica con dati partizionati su un massimo di 500 partizioni.

Ogni shard in un cluster dispone di un singolo nodo primario lettura/scrittura e fino a 5 nodi replica di sola lettura. Ogni nodo primario può sostenere fino a 100 MB/s. Puoi creare un cluster con un numero più alto partizioni e un numero più basso di repliche per un totale di 500 nodi per cluster. Questa configurazione del cluster può andare da 500 partizioni e 0 repliche a 100 partizioni e 4 repliche che è il numero massimo consentito di repliche.

Consistenza

In MemoryDB, i nodi primari sono fortemente coerenti. Le operazioni di scrittura riuscite vengono archiviate in modo duraturo in registri transazionali Multi-AZ distribuiti prima di essere restituiti ai client. Le operazioni di lettura sui file primari restituiscono sempre la maggior parte up-to-date dei dati che riflettono gli effetti di tutte le precedenti operazioni di scrittura riuscite. Questa solida coerenza viene preservata nei failover primari.

In MemoryDB, i nodi replica sono a consistenza finale. Le operazioni di lettura dalle repliche (utilizzando `READONLY` il comando) potrebbero non sempre riflettere gli effetti delle più recenti operazioni di scrittura riuscite, con le metriche del ritardo pubblicate su `CloudWatch`. Tuttavia, le operazioni di lettura da una singola replica sono sequenzialmente coerenti. Le operazioni di scrittura riuscite hanno effetto su ogni replica nello stesso ordine in cui sono state eseguite sulla replica primaria.

Replica in un cluster

Ogni replica letta in uno shard conserva una copia dei dati dal nodo primario dello shard. I meccanismi di replica asincrona che utilizzano i registri delle transazioni vengono utilizzati per mantenere le repliche lette sincronizzate con quelle primarie. Le applicazioni possono leggere da qualsiasi nodo nel cluster. Le applicazioni possono scrivere solo nei nodi primari. Le repliche di lettura migliorano la scalabilità di lettura. Poiché MemoryDB archivia i dati in registri delle transazioni durevoli, non vi è alcun rischio che i dati vadano persi. I dati vengono partizionati tra gli shard in un cluster MemoryDB.

Le applicazioni utilizzano l'endpoint del cluster MemoryDB per connettersi con i nodi del cluster. Per ulteriori informazioni, consulta [Individuazione degli endpoint di connessione](#).

I cluster MemoryDB sono regionali e possono contenere nodi di una sola regione. Per migliorare la tolleranza agli errori, è necessario fornire le repliche primarie e leggere le repliche su più zone di disponibilità all'interno di quella regione.

L'uso della replica, che fornisce Multi-AZ, è fortemente consigliato per tutti i cluster MemoryDB. Per ulteriori informazioni, consulta [Riduzione al minimo dei tempi di inattività in MemoryDB con Multi-AZ](#).

Riduzione al minimo dei tempi di inattività in MemoryDB con Multi-AZ

Esistono diversi casi in cui MemoryDB potrebbe dover sostituire un nodo primario; questi includono alcuni tipi di manutenzione pianificata e l'improbabile evento di un guasto di un nodo primario o di una zona di disponibilità.

La risposta all'errore del nodo dipende dal nodo che ha avuto esito negativo. Tuttavia, in tutti i casi, MemoryDB garantisce che nessun dato venga perso durante la sostituzione o il failover dei nodi. Ad esempio, se una replica fallisce, il nodo fallito viene sostituito e i dati vengono sincronizzati dal registro delle transazioni. In caso di errore del nodo primario, viene attivato un failover su una replica coerente che garantisce l'assenza di perdita di dati durante il failover. Le scritture vengono ora eseguite dal nuovo nodo primario. Il vecchio nodo primario viene quindi sostituito e sincronizzato dal registro delle transazioni.

Se un nodo primario fallisce su uno shard a nodo singolo (nessuna replica), MemoryDB smette di accettare le scritture fino a quando il nodo primario non viene sostituito e sincronizzato dal registro delle transazioni.

La sostituzione dei nodi può comportare alcuni tempi di inattività per il cluster, ma se Multi-AZ è attivo, i tempi di inattività sono ridotti al minimo. Il ruolo del nodo primario eseguirà automaticamente il failover su una delle repliche. Non è necessario creare e fornire un nuovo nodo primario, poiché MemoryDB lo gestirà in modo trasparente. Questo failover e la promozione delle repliche garantiscono la possibilità di ricominciare a scrivere nel nuovo nodo primario non appena la promozione è terminata.

In caso di sostituzioni pianificate dei nodi avviate a causa di aggiornamenti di manutenzione o aggiornamenti del servizio, tieni presente che le sostituzioni pianificate dei nodi vengono completate mentre il cluster soddisfa le richieste di scrittura in arrivo.

La tecnologia Multi-AZ sui cluster MemoryDB migliora la tolleranza ai guasti. Questo è vero in particolare nei casi in cui i nodi primari del cluster diventano irraggiungibili o falliscono per qualsiasi motivo. Multi-AZ sui cluster MemoryDB richiede che ogni shard abbia più di un nodo e viene abilitato automaticamente.

Argomenti

- [Risposte per scenari di errore relativi alla funzione Multi-AZ](#)
- [Test del failover automatico](#)

Risposte per scenari di errore relativi alla funzione Multi-AZ

Se Multi-AZ è attivo, un nodo primario fallito esegue il failover su una replica disponibile. La replica viene sincronizzata automaticamente con il registro delle transazioni e diventa primaria, il che è molto più veloce rispetto alla creazione e al riprovisioning di un nuovo nodo primario. Questo processo richiede in genere pochi secondi prima che sia possibile scrivere nuovamente nel cluster.

Quando Multi-AZ è attivo, MemoryDB monitora continuamente lo stato del nodo primario. Se il nodo primario non riesce, viene eseguita una delle seguenti operazioni a seconda del tipo di errore.

Argomenti

- [Scenari di errore quando solo il nodo primario non riesce](#)
- [Scenari di errore in caso di errore del nodo primario e di alcune repliche](#)
- [Scenari di fallimento quando l'intero cluster non riesce](#)

Scenari di errore quando solo il nodo primario non riesce

Se si verifica un errore solo nel nodo primario, una replica diventerà automaticamente primaria. Una replica sostitutiva viene quindi creata e fornita nella stessa zona di disponibilità della replica primaria fallita.

Quando fallisce solo il nodo primario, MemoryDB Multi-AZ esegue le seguenti operazioni:

1. Il nodo primario non riuscito viene portato offline.
2. Una up-to-date replica diventa automaticamente primaria.

Le scritture possono riprendere non appena il processo di failover è completo, in genere solo pochi secondi.

3. Viene avviata e fornita una replica sostitutiva.

La replica sostitutiva viene avviata nella zona di disponibilità in cui si trovava il nodo primario guasto in modo da mantenere la distribuzione dei nodi.

4. La replica si sincronizza con il registro delle transazioni.

Per informazioni sull'individuazione degli endpoint di un cluster, consulta i seguenti argomenti:

- [Ricerca dell'endpoint per un cluster MemoryDB \(API MemoryDB\)](#)

Scenari di errore in caso di errore del nodo primario e di alcune repliche

Se la replica primaria e almeno una replica falliscono, una up-to-date replica viene promossa a cluster primario. Le nuove repliche vengono inoltre create e fornite nelle stesse zone di disponibilità dei nodi guasti.

Quando il nodo primario e alcune repliche falliscono, MemoryDB Multi-AZ esegue le seguenti operazioni:

1. Il nodo primario fallito e le repliche non riuscite vengono messi in modalità offline.
2. Una replica disponibile diventerà il nodo principale.

Le scritture possono riprendere non appena il failover è completo, in genere solo pochi secondi.

3. Repliche sostitutive vengono create e sottoposte a provisioning.

Le repliche sostitutive vengono create nelle zone di disponibilità dei nodi non riusciti, in modo da mantenere la distribuzione dei nodi.

4. Tutti i nodi si sincronizzano con il registro delle transazioni.

Per informazioni sull'individuazione degli endpoint di un cluster, consulta i seguenti argomenti:

- [Ricerca dell'endpoint per un cluster MemoryDB \(AWSCLIP\)](#)
- [Ricerca dell'endpoint per un cluster MemoryDB \(API MemoryDB\)](#)

Scenari di fallimento quando l'intero cluster non riesce

In caso di errore generale, tutti i nodi vengono ricreati e sottoposti a provisioning nelle stesse zone di disponibilità dei nodi originali.

Non vi è alcuna perdita di dati in questo scenario poiché i dati sono stati mantenuti nel registro delle transazioni.

Quando l'intero cluster fallisce, MemoryDB Multi-AZ esegue le seguenti operazioni:

1. Il nodo primario e le repliche non funzionanti vengono messi offline.

2. Viene creato e fornito un nodo primario sostitutivo, sincronizzato con il registro delle transazioni.
3. Le repliche sostitutive vengono create e assegnate, sincronizzate con il registro delle transazioni.

Le sostituzioni vengono create nelle zone di disponibilità dei nodi non riusciti, in modo da mantenere la distribuzione dei nodi.

Per informazioni sull'individuazione degli endpoint di un cluster, consulta i seguenti argomenti:

- [Ricerca dell'endpoint per un cluster MemoryDB \(AWSCLIP\)](#)
- [Ricerca dell'endpoint per un cluster MemoryDB \(API MemoryDB\)](#)

Test del failover automatico

È possibile testare il failover automatico utilizzando la console MemoryDBAWS CLI, l'API e MemoryDB.

Durante il test, tieni presente quanto segue:

- È possibile utilizzare questa operazione fino a cinque volte in un periodo di 24 ore.
- Se richiami questa operazione su shard in cluster diversi, puoi effettuare le chiamate contemporaneamente.
- In alcuni casi, è possibile chiamare questa operazione più volte su shard diversi nello stesso cluster MemoryDB. In questi casi, la sostituzione del primo nodo deve essere completata prima di effettuare una chiamata successiva.
- Per determinare se la sostituzione del nodo è completa, controlla gli eventi utilizzando la console MemoryDB for RedisAWS CLI, l'API o MemoryDB. Cerca i seguenti eventi correlati a `FailoverShard`, elencati qui in ordine di probabile evento:
 1. messaggio del cluster: `FailoverShard API called for shard <shard-id>`
 2. messaggio del cluster: `Failover from primary node <primary-node-id> to replica node <node-id> completed`
 3. messaggio del cluster: `Recovering nodes <node-id>`
 4. messaggio del cluster: `Finished recovery for nodes <node-id>`

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [DescribeEvents](#) nel riferimento dell'API di MemoryDB
- Questa API è progettata per testare il comportamento dell'applicazione in caso di failover di MemoryDB. Non è progettato per essere uno strumento operativo per l'avvio di un failover per risolvere un problema con il cluster. Inoltre, in determinate condizioni come eventi operativi su larga scala, AWS potrebbe bloccare questa API.

Argomenti

- [Test del failover automatico utilizzando la AWS Management Console](#)
- [Test del failover automatico utilizzando la AWS CLI](#)
- [Test del failover automatico utilizzando l'API MemoryDB](#)

Test del failover automatico utilizzando la AWS Management Console

Utilizza la procedura seguente per testare il failover automatico con la console.

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. Scegli il pulsante radio a sinistra del cluster che desideri testare. Questo cluster deve avere almeno un nodo di replica.
3. Nell'area Dettagli, conferma che questo cluster è abilitato per Multi-AZ. Se il cluster non è abilitato per la funzione Multi-AZ, scegliere un cluster diverso o modificare questo cluster per abilitare la funzione Multi-AZ. Per ulteriori informazioni, consulta [Modifica di un cluster MemoryDB](#).
4. Seleziona il nome del cluster.
5. Nella pagina Frammenti e nodi, scegli il nome dello shard su cui desideri testare il failover.
6. Per il nodo, scegli Failover Primary.
7. Scegli Continua per eseguire il failover nel nodo primario o Annulla per annullare l'operazione e non eseguire il failover nel nodo primario.

Durante il processo di failover, la console continua a visualizzare lo stato del nodo come disponibile. Per monitorare l'avanzamento del test di failover, scegli Eventi dal riquadro di navigazione della console. Nella scheda Eventi, cerca gli eventi che indicano che il failover è stato avviato (`FailoverShard API called`) e completato (`Recovery completed`).

Test del failover automatico utilizzando la AWS CLI

[È possibile testare il failover automatico su qualsiasi cluster compatibile con Multi-AZ utilizzando l'AWS CLI operazione failover-shard.](#)

Parametri

- `--cluster-name` - Obbligatorio Il cluster che deve essere testato.
- `--shard-name` - Obbligatorio Il nome dello shard su cui si desidera testare il failover automatico. Puoi testare un massimo di cinque frammenti in un periodo di 24 ore consecutive.

L'esempio seguente utilizza AWS CLI per `failover-shard` richiamare lo shard `0001` nel cluster MemoryDB. `my-cluster`

Per Linux, macOS o Unix:

```
aws memorydb failover-shard \  
  --cluster-name my-cluster \  
  --shard-name 0001
```

Per Windows:

```
aws memorydb failover-shard ^  
  --cluster-name my-cluster ^  
  --shard-name 0001
```

Per monitorare l'avanzamento del failover, utilizza l'operazione `describe-events` di AWS CLI.

Restituirà la seguente risposta 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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [condivisione tramite failover](#)
- [describe-events](#)

Test del failover automatico utilizzando l'API MemoryDB

L'esempio seguente richiama `FailoverShard` lo shard `0003` nel cluster `memorydb00`.

Example Test del failover automatico

```
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>
```

Per monitorare l'avanzamento del failover, utilizza l'operazione dell'API `MemoryDBDescribeEvents`.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [FailoverShard](#)
- [DescribeEvents](#)

Modifica del numero di repliche

Puoi aumentare o diminuire dinamicamente il numero di repliche di replica nel cluster MemoryDB utilizzando l'API AWS Management Console, la o la MemoryDB. AWS CLI Tutti i frammenti devono avere lo stesso numero di repliche.

Incremento del numero di repliche in un cluster

Puoi aumentare il numero di repliche in un cluster MemoryDB fino a un massimo di cinque per shard. A questo scopo, utilizza lo AWS Management Console, la o l'AWS CLI API Memorydatabase.

Argomenti

- [Utilizzo di AWS Management Console](#)
- [Utilizzo di AWS CLI](#)
- [Utilizzo dell'API Memorydatabase](#)

Utilizzo di AWS Management Console

Per aumentare il numero di repliche in un cluster MemoryDB (console), vedere. [Aggiunta/rimozione di nodi da un cluster](#)

Utilizzo di AWS CLI

Per aumentare il numero di repliche in un cluster MemoryDB, utilizzare il `update-cluster` comando dell'con i parametri seguenti:

- `--cluster-name` - Obbligatorio Indica in quale cluster si desidera aumentare il numero di repliche.
- `--replica-configuration` - Obbligatorio Consente di impostare il numero di repliche. Per aumentare il numero di repliche, impostate la `ReplicaCount` proprietà sul numero di repliche desiderato in questo shard al termine di questa operazione.

Example

L'esempio seguente incrementa il numero di repliche nel cluster `my-cluster` a 2.

Per Linux, macOS o Unix:

```
aws memorydb update-cluster \  
  --cluster-name my-cluster \  
  --replica-configuration \  
    ReplicaCount=2
```

Per Windows:


```
aws memorydb update-cluster ^
  --cluster-name my-cluster ^
  --replica-configuration ^
    ReplicaCount=2
```

Restituisce la risposta JSON seguente:

```
{
  "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
  }
}
```

Per visualizzare i dettagli del cluster aggiornato una volta che il suo stato passa da aggiornamento a disponibile, usa il seguente comando:

Per Linux, macOS o Unix:

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

Per Windows:

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

Restituirà la seguente risposta 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
}
]
}

```

Per ulteriori informazioni sull'aumento del numero di repliche mediante la CLI, consulta [update-cluster](#) nella guida di riferimento ai comandi. AWS CLI

Utilizzo dell'API Memorydatabase

Per aumentare il numero di repliche in uno shard di MemoryDB, utilizzate l'UpdateClusterazione con i seguenti parametri:

- `ClusterName` - Obbligatorio Indica in quale cluster si desidera aumentare il numero di repliche.
- `ReplicaConfiguration` - Obbligatorio Consente di impostare il numero di repliche. Per aumentare il numero di repliche, impostate la `ReplicaCount` proprietà sul numero di repliche desiderato in questo shard al termine di questa operazione.

Example

L'esempio seguente incrementa il numero di repliche nel cluster `sample-cluster` a tre. Quando l'esempio è finito, ci sono tre repliche in ogni frammento. Questo numero si applica sia che si tratti di un cluster MemoryDB con un singolo shard o di un cluster MemoryDB con più shard.

```
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>
```

Per ulteriori informazioni sull'aumento del numero di repliche mediante l'API, consulta [UpdateCluster](#)

Riduzione del numero di repliche in un cluster

Puoi ridurre il numero di repliche in un cluster per MemoryDB. Puoi ridurre il numero di repliche a zero, ma non puoi eseguire il failover su una replica in caso di errore del nodo primario.

Puoi utilizzare laAWS Management Console, la AWS CLI o l'API Memorydatabase per ridurre il numero di repliche in un cluster.

Argomenti

- [Utilizzo di AWS Management Console](#)
- [Utilizzo di AWS CLI](#)
- [Utilizzo dell'API Memorydatabase](#)

Utilizzo di AWS Management Console

Per ridurre il numero di repliche in un cluster MemoryDB (console), vedere. [Aggiunta/rimozione di nodi da un cluster](#)

Utilizzo di AWS CLI

Per ridurre il numero di repliche in un cluster MemoryDB, utilizzare il `update-cluster` comando dell'`con` i parametri seguenti:

- `--cluster-name` - Obbligatorio Identifica in quale cluster si desidera ridurre il numero di repliche.
- `--replica-configuration` - Obbligatorio

`ReplicaCount`— Imposta questa proprietà per specificare il numero nodi nodi replica desiderati.

Example

L'esempio seguente viene utilizzato `--replica-configuration` per ridurre il numero di repliche nel cluster `my-cluster` al valore specificato.

Per Linux, macOS o Unix:

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

Per Windows:

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

Restituirà la seguente risposta 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
  }
}
```

Per visualizzare i dettagli del cluster aggiornato una volta che il suo stato passa da aggiornamento a disponibile, usa il seguente comando:

Per Linux, macOS o Unix:

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

Per Windows:

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

Restituirà la seguente risposta 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
}
]
}

```

Per ulteriori informazioni sulla riduzione del numero di repliche mediante la CLI, consulta [update-cluster](#) nella guida di riferimento ai comandi. AWS CLI

Utilizzo dell'API Memorydatabase

Per ridurre il numero di repliche in un cluster MemoryDB, utilizzare l'UpdateClusterazione con i seguenti parametri:

- **ClusterName** - Obbligatorio Identifica in quale cluster si desidera ridurre il numero di repliche.
- **ReplicaConfiguration** - Obbligatorio Consente di impostare il numero di repliche.

ReplicaCount— Imposta questa proprietà per specificare il numero di nodi replica desiderati.

Example

L'esempio seguente viene utilizzato `ReplicaCount` per ridurre il numero di repliche nel cluster `sample-cluster` a uno. Quando l'esempio è finito, c'è una replica in ogni frammento. Questo numero si applica sia che si tratti di un cluster MemoryDB con un singolo shard o di un cluster MemoryDB con più shard.

```
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>
```

Per ulteriori informazioni sulla riduzione del numero di repliche mediante l'API, consulta.

[UpdateCluster](#)

Snapshot e ripristino

I cluster MemoryDB for Redis eseguono automaticamente il backup dei dati in un log transazionale Multi-AZ, ma puoi scegliere di creare istantanee di un cluster periodicamente o su richiesta. point-in-time Queste istantanee possono essere utilizzate per ricreare un cluster in un punto precedente o per creare un cluster nuovo di zecca. L'istantanea è costituita dai metadati del cluster, insieme a tutti i dati del cluster. Tutte le istantanee vengono scritte su Amazon Simple Storage Service (Amazon S3), che fornisce uno storage durevole. In qualsiasi momento, puoi ripristinare i dati creando un nuovo cluster MemoryDB e popolandolo con i dati di uno snapshot. Con MemoryDB, è possibile gestire le istantanee utilizzando l'API AWS Management Console, the AWS Command Line Interface (AWS CLI) e MemoryDB.

Argomenti

- [Vincoli relativi alle istantanee](#)
- [Costi delle istantanee](#)
- [Pianificazione di istantanee automatiche](#)
- [Creazione di istantanee manuali](#)
- [Creazione di un'istantanea finale](#)

- [Descrizione delle istantanee](#)
- [Copia di uno snapshot](#)
- [Esportazione di un'istananea](#)
- [Ripristino da uno snapshot](#)
- [Seminare un nuovo cluster con un'istananea creata esternamente](#)
- [Taggare le istantanee](#)
- [Eliminazione di uno snapshot](#)

Vincoli relativi alle istantanee

Considerate i seguenti vincoli durante la pianificazione o la creazione di istantanee:

- Per i cluster MemoryDB, snapshot e restore sono disponibili per tutti i tipi di nodi supportati.
- Durante un periodo contiguo di 24 ore, è possibile creare non più di 20 istantanee manuali per cluster.
- MemoryDB supporta solo l'acquisizione di istantanee a livello di cluster. MemoryDB non supporta l'acquisizione di istantanee a livello di shard o di nodo.
- Durante il processo di snapshot, non puoi eseguire altre operazioni API o CLI sul cluster.
- Se si elimina un cluster e si richiede un'istananea finale, MemoryDB acquisisce sempre l'istananea dai nodi primari. Ciò garantisce l'acquisizione dei dati più recenti prima dell'eliminazione del cluster.

Costi delle istantanee

Utilizzando MemoryDB, è possibile archiviare gratuitamente un'istananea per ogni cluster MemoryDB attivo. Lo spazio di archiviazione per istantanee aggiuntive viene addebitato a una tariffa di 0,085 USD/GB al mese per tutte le regioni. AWS Non sono previsti costi di trasferimento dei dati per la creazione di un'istananea o per il ripristino dei dati da un'istananea a un cluster MemoryDB.

Pianificazione di istantanee automatiche

Per qualsiasi cluster MemoryDB, è possibile abilitare le istantanee automatiche. Quando le istantanee automatiche sono abilitate, MemoryDB crea un'istananea del cluster su base giornaliera. Non vi è alcun impatto sul cluster e la modifica è immediata. Per ulteriori informazioni, consulta [Ripristino da uno snapshot](#).

Quando si pianificano istantanee automatiche, è necessario pianificare le seguenti impostazioni:

- Finestra istantanea: un periodo durante ogni giorno in cui MemoryDB inizia a creare un'istananea. La durata minima per la finestra delle istantanee è di 60 minuti. È possibile impostare la finestra delle istantanee in qualsiasi momento, quando lo ritieni più comodo, o per un'ora del giorno che eviti di creare istantanee durante periodi di utilizzo particolarmente intenso.

Se non si specifica una finestra di istantanea, MemoryDB ne assegna una automaticamente.

- Limite di conservazione degli snapshot: il numero di giorni in cui lo snapshot viene conservato in Amazon S3. Ad esempio, se imposti il limite di conservazione su 5, una istantanea scattata oggi viene conservata per 5 giorni. Quando il limite di conservazione scade, l'istananea viene eliminata automaticamente.

Il limite massimo di conservazione delle istantanee è di 35 giorni. Se il limite di conservazione delle istantanee è impostato su 0, le istantanee automatiche sono disabilitate per il cluster. I dati di MemoryDB sono ancora completamente durevoli anche con le istantanee automatiche disattivate.

È possibile abilitare o disabilitare le istantanee automatiche durante la creazione di un cluster MemoryDB utilizzando la console MemoryDB, l'API MemoryDB o l'API MemoryDB. AWS CLI È possibile abilitare le istantanee automatiche quando si crea un cluster MemoryDB selezionando la casella Abilita backup automatici nella sezione Istananee. Per ulteriori informazioni, consulta [Creazione di un cluster MemoryDB](#).

Creazione di istantanee manuali

Oltre alle istantanee automatiche, è possibile creare un'istananea manuale in qualsiasi momento. A differenza delle istantanee automatiche, che vengono eliminate automaticamente dopo un periodo di conservazione specificato, le istantanee manuali non hanno un periodo di conservazione dopo il quale vengono eliminate automaticamente. È necessario eliminare manualmente qualsiasi istantanea manuale. Anche se si elimina un cluster o un nodo, tutte le istantanee manuali di quel cluster o nodo vengono conservate. Se non desideri più conservare un'istananea manuale, devi eliminarla tu stesso in modo esplicito.

Le istantanee manuali sono utili per il test e l'archiviazione. Supponi ad esempio di aver sviluppato un set di dati di riferimento per scopi di test. Puoi creare un'istananea manuale dei dati e ripristinarla quando vuoi. Dopo aver testato un'applicazione che modifica i dati, è possibile reimpostare i dati creando un nuovo cluster ed eseguendo il ripristino dallo snapshot di base. Quando il cluster è pronto, è possibile testare nuovamente le applicazioni rispetto ai dati di base e ripetere questa procedura con la frequenza necessaria.

Oltre a creare direttamente un'istananea manuale, è possibile creare un'istananea manuale in uno dei seguenti modi:

- [Copia di uno snapshot](#)— Non importa se l'istananea di origine è stata creata automaticamente o manualmente.
- [Creazione di un'istananea finale](#)— Crea un'istananea immediatamente prima di eliminare un cluster.

Altri argomenti importanti

- [Vincoli relativi alle istantanee](#)
- [Costi delle istantanee](#)

È possibile creare un'istananea manuale di un nodo utilizzando l'API AWS Management Console MemoryDB o l' AWS CLI API MemoryDB.

Creazione di un'istantanea manuale (Console)

Per creare un'istantanea di un cluster (console)

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. dal riquadro di navigazione a sinistra, scegli Clusters.

Viene visualizzata la schermata dei cluster di MemoryDB.

3. scegli il pulsante di opzione a sinistra del nome del cluster MemoryDB di cui desideri eseguire il backup.
4. Scegli Azioni e poi Scatta un'istantanea.
5. Nella finestra Istantanea, digita un nome per l'istantanea nella casella Nome istantanea. È consigliabile che il nome indichi il cluster di cui è stato eseguito il backup e la data e l'ora in cui è stata creata l'istantanea.

I vincoli di denominazione dei cluster sono i seguenti:

- Devono contenere da 1 a 40 caratteri alfanumerici o trattini.
 - Devono iniziare con una lettera.
 - Non possono contenere due trattini consecutivi.
 - Non possono terminare con un trattino.
6. In Crittografia, scegli se utilizzare una chiave di crittografia predefinita o una chiave gestita dal cliente. Per ulteriori informazioni, consulta [Crittografia in transito \(TLS\) in MemoryDB](#).
 7. In Tag, puoi aggiungere opzionalmente tag per cercare e filtrare le istantanee o tenere traccia AWS dei costi.
 8. Seleziona Acquisisci snapshot.

Lo stato de cluster cambia in creazione di snapshot. Quando lo stato torna disponibile, l'istantanea è completa.

Creazione di un'istantanea manuale (AWS CLI)

Per creare un'istantanea manuale di un cluster utilizzando il AWS CLI, utilizzare l'create-snapshot AWS CLI operazione con i seguenti parametri:

- `--cluster-name`— Nome del cluster MemoryDB da utilizzare come origine per l'istantanea. Utilizzate questo parametro per il backup di un cluster MemoryDB.

I vincoli di denominazione dei cluster sono i seguenti:

- Devono contenere da 1 a 40 caratteri alfanumerici o trattini.
 - Devono iniziare con una lettera.
 - Non possono contenere due trattini consecutivi.
 - Non possono terminare con un trattino.
-
- `--snapshot-name` - Nome dello snapshot da creare.

Argomenti correlati

Per ulteriori informazioni, consulta la sezione `create-snapshot` nella Documentazione di riferimento della AWS CLI .

Creazione di un'istantanea manuale (API MemoryDB)

Per creare un'istantanea manuale di un cluster utilizzando l'API MemoryDB, utilizzate l'operazione `API CreateSnapshot MemoryDB` con i seguenti parametri:

- `ClusterName`— Nome del cluster MemoryDB da utilizzare come origine per l'istantanea. Utilizzate questo parametro per il backup di un cluster MemoryDB.

I vincoli di denominazione dei cluster sono i seguenti:

- Devono contenere da 1 a 40 caratteri alfanumerici o trattini.
 - Devono iniziare con una lettera.
 - Non possono contenere due trattini consecutivi.
 - Non possono terminare con un trattino.
-
- `SnapshotName` - Nome dello snapshot da creare.

Argomenti correlati

Per ulteriori informazioni, vedere. [CreateSnapshot](#)

Creazione di un'istantanea finale

È possibile creare un'istantanea finale utilizzando la console MemoryDB AWS CLI, o l'API MemoryDB.

Creazione di un'istantanea finale (Console)

È possibile creare un'istantanea finale quando si elimina un cluster MemoryDB utilizzando la console MemoryDB.

Per creare un'istantanea finale quando si elimina un cluster di MemoryDB, nella pagina di eliminazione, scegli Sì e assegna un nome all'istantanea in. [Fase 4: Eliminazione di un cluster](#)

Creazione di un'istantanea finale (AWS CLI)

È possibile creare un'istantanea finale quando si elimina un cluster MemoryDB utilizzando il. AWS CLI

Quando si elimina un cluster MemoryDB

Per creare un'istantanea finale quando si elimina un cluster, utilizzate l'`delete-cluster` AWS CLI operazione con i seguenti parametri:

- `--cluster-name` : Nome del cluster in corso di eliminazione.
- `--final-snapshot-name`— Nome dell'istantanea finale.

Il codice seguente scatta l'istantanea finale `bkup-20210515-final` quando si elimina il cluster. `myCluster`

Per Linux, macOS o Unix:

```
aws memorydb delete-cluster \  
  --cluster-name myCluster \  
  --final-snapshot-name bkup-20210515-final
```

Per Windows:

```
aws memorydb delete-cluster ^  
  --cluster-name myCluster ^  
  --final-snapshot-name bkup-20210515-final
```

Per ulteriori informazioni, vedere [delete-cluster](#) nel Command Reference.AWS CLI

Creazione di un'istantanea finale (API MemoryDB)

È possibile creare un'istantanea finale quando si elimina un cluster MemoryDB utilizzando l'API MemoryDB.

Quando si elimina un cluster MemoryDB

Per creare un'istantanea finale, utilizzate l'operazione API `DeleteCluster` MemoryDB con i seguenti parametri.

- `ClusterName` : Nome del cluster in corso di eliminazione.
- `FinalSnapshotName`— Nome dell'istantanea.

La seguente operazione dell'API MemoryDB crea l'istantanea `bkup-20210515-final` durante l'eliminazione del cluster `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>
```

Per ulteriori informazioni, vedere. [DeleteCluster](#)

Descrizione delle istantanee

Le seguenti procedure mostrano come visualizzare un elenco delle istantanee. Se lo desideri, puoi anche visualizzare i dettagli di una particolare istantanea.

Descrizione delle istantanee (Console)

Per visualizzare le istantanee utilizzando il AWS Management Console

1. Accedere alla console
2. dal riquadro di navigazione a sinistra, scegli Istantanee.
3. Usa la ricerca per filtrare le istantanee manuali, automatiche o tutte le istantanee.
4. Per visualizzare i dettagli di una particolare istantanea, scegli il pulsante di opzione a sinistra del nome dell'istantanea. Scegli Azioni, quindi Visualizza dettagli.
5. Facoltativamente, nella pagina Visualizza dettagli, puoi eseguire ulteriori azioni di istantanea come copiare, ripristinare o eliminare. È inoltre possibile aggiungere tag all'istantanea

Descrizione delle istantanee (CLI AWS)

Per visualizzare un elenco di istantanee e, facoltativamente, dettagli su un'istantanea specifica, utilizzate l'operazione `describe-snapshots` CLI.

Examples (Esempi)

La seguente operazione utilizza il parametro `--max-results` per elencare fino a 20 istantanee associate all'account. L'omissione del parametro `--max-results` elenca fino a 50 istantanee.

```
aws memorydb describe-snapshots --max-results 20
```

La seguente operazione utilizza il parametro `--cluster-name` per elencare solo le istantanee associate al cluster. `my-cluster`

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

L'operazione seguente utilizza il parametro `--snapshot-name` per visualizzare i dettagli dell'istantanea `my-snapshot`.

```
aws memorydb describe-snapshots --snapshot-name my-snapshot
```

Per ulteriori informazioni, vedere [describe-snapshots](#).

Descrizione delle istantanee (API MemoryDB)

Per visualizzare un elenco di istantanee, utilizzare l'operazione. `DescribeSnapshots`

Examples (Esempi)

La seguente operazione utilizza il parametro `MaxResults` per elencare fino a 20 istantanee associate all'account. L'omissione del parametro `MaxResults` elenca fino a 50 istantanee.

```
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 seguente operazione utilizza il parametro `ClusterName` per elencare tutte le istantanee associate al cluster. `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>
```

L'operazione seguente utilizza il parametro `SnapshotName` per visualizzare i dettagli dell'istantanea `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>
```

Per ulteriori informazioni, vedere [DescribeSnapshots](#).

Copia di uno snapshot

È possibile creare una copia di qualsiasi istantanea, indipendentemente dal fatto che sia stata creata automaticamente o manualmente. Quando si copia un'istantanea, per la destinazione viene utilizzata la stessa chiave di crittografia KMS dell'origine, a meno che non venga specificatamente sovrascritta. Puoi anche esportare la tua istantanea in modo da poterti accedere dall'esterno di MemoryDB. Per indicazioni sull'esportazione dell'istantanea, consulta [Esportazione di un'istantanea](#)

Le seguenti procedure mostrano come copiare un'istantanea.

Copiare un'istantanea (Console)

Per copiare un'istantanea (console)

1. [Accedere AWS Management Console e aprire la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Per visualizzare un elenco delle tue istantanee, dal riquadro di navigazione a sinistra scegli Istanee.
3. Dall'elenco delle istantanee, scegli il pulsante di opzione a sinistra del nome dell'istantanea che desideri copiare.
4. Scegli Azioni, quindi scegli Copia.
5. Nella pagina Copia istantanea, procedi come segue:
 - a. Nella casella Nome nuova istantanea, digitate un nome per la nuova istantanea.
 - b. Lasciare vuota la casella Target S3 Bucket (Bucket S3 di destinazione) opzionale. Questo campo deve essere utilizzato solo per esportare l'istantanea e richiede autorizzazioni S3 speciali. Per informazioni sull'esportazione di un'istantanea, consulta [Esportazione di un'istantanea](#)
 - c. Scegli se utilizzare la chiave di AWS KMS crittografia predefinita o utilizzare una chiave personalizzata. Per ulteriori informazioni, consulta [Crittografia in transito \(TLS\) in MemoryDB.](#)
 - d. Facoltativamente, puoi anche aggiungere tag alla copia dell'istantanea.
 - e. Scegli Copia.

Copiare un'istantanea (CLI AWS)

Per copiare un'istantanea, utilizzare l'operazione. `copy-snapshot`

Parametri

- `--source-snapshot-name`— Nome dell'istantanea da copiare.
- `--target-snapshot-name`— Nome della copia dell'istantanea.
- `--target-bucket`— Riservato all'esportazione di un'istantanea. Non utilizzare questo parametro quando si crea una copia di un'istantanea. Per ulteriori informazioni, consulta [Esportazione di un'istantanea](#).

L'esempio seguente crea una copia di un'istantanea automatica.

Per 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
```

Per Windows:

```
aws memorydb copy-snapshot ^  
  --source-snapshot-name automatic.my-primary-2021-03-27-03-15 ^  
  --target-snapshot-name my-snapshot-copy
```

Per ulteriori informazioni, vedere [copy-snapshot](#).

Copiare un'istantanea (API MemoryDB)

Per copiare un'istantanea, utilizzate l'`copy-snapshot` operazione con i seguenti parametri:

Parametri

- `SourceSnapshotName`— Nome dell'istantanea da copiare.
- `TargetSnapshotName`— Nome della copia dell'istantanea.
- `TargetBucket`— Riservato all'esportazione di un'istantanea. Non utilizzare questo parametro quando si crea una copia di un'istantanea. Per ulteriori informazioni, consulta [Esportazione di un'istantanea](#).

L'esempio seguente crea una copia di un'istantanea automatica.

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>
```

Per ulteriori informazioni, vedere [CopySnapshot](#).

Esportazione di un'istantanea

MemoryDB for Redis supporta l'esportazione dello snapshot di MemoryDB in un bucket Amazon Simple Storage Service (Amazon S3), che consente di accedervi dall'esterno di MemoryDB. Le istantanee di MemoryDB esportate sono completamente compatibili con Redis open source e possono essere caricate con la versione o gli strumenti Redis appropriati. È possibile esportare un'istantanea utilizzando la console MemoryDB, o l'API MemoryDB. AWS CLI

L'esportazione di un'istantanea può essere utile se è necessario avviare un cluster in un'altra regione. AWS È possibile esportare i dati in una AWS regione, copiare il file.rdb AWS nella nuova regione e quindi utilizzare il file.rdb per eseguire il seeding del nuovo cluster anziché attendere che il nuovo cluster venga popolato tramite l'uso. Per informazioni sull'inizializzazione di un nuovo cluster, consulta [Seminare un nuovo cluster con un'istantanea creata esternamente](#). Un altro motivo per espandere i dati del cluster potrebbe essere per utilizzare il file .rdb per l'elaborazione offline.

Important

- Lo snapshot di MemoryDB e il bucket Amazon S3 in cui desideri copiarlo devono trovarsi nella stessa regione. AWS

Sebbene le istantanee copiate in un bucket Amazon S3 siano crittografate, ti consigliamo vivamente di non concedere ad altri l'accesso al bucket Amazon S3 in cui desideri archiviare le tue istantanee.

- L'esportazione di uno snapshot in Amazon S3 non è supportata per i cluster che utilizzano il tiering dei dati. Per ulteriori informazioni, consulta [Tiering di dati](#).

Prima di poter esportare uno snapshot in un bucket Amazon S3, devi avere un bucket Amazon S3 nella stessa regione dello snapshot. AWS Concedi a MemoryDB l'accesso al bucket. Le prime due fasi mostrano come eseguire questa operazione.

Warning

Nei seguenti scenari i dati potrebbero essere esposti in modi indesiderati:

- Quando un'altra persona ha accesso al bucket Amazon S3 in cui hai esportato la tua istantanea.

Per controllare l'accesso alle tue istantanee, consenti l'accesso al bucket Amazon S3 solo a coloro a cui desideri accedere ai tuoi dati. Per informazioni sulla gestione dell'accesso utente ai bucket Amazon S3, consultare [Controllo degli accessi](#) nella Guida per gli sviluppatori di Amazon S3.

- Quando un'altra persona dispone delle autorizzazioni per utilizzare l'operazione API. CopySnapshot

Gli utenti o i gruppi che dispongono delle autorizzazioni per utilizzare l'operazione CopySnapshot API possono creare i propri bucket Amazon S3 e copiarvi le istantanee. Per controllare l'accesso alle tue istantanee, utilizza una policy AWS Identity and Access Management (IAM) per controllare chi è in grado di utilizzare l'API. CopySnapshot Per ulteriori informazioni sull'utilizzo di IAM per controllare l'uso delle operazioni dell'API MemoryDB, consulta [Gestione delle identità e degli accessi in MemoryDB per Redis](#) la Guida per l'utente di MemoryDB.

Argomenti

- [Fase 1: creazione di un bucket Amazon S3](#)
- [Fase 2: concedere a MemoryDB l'accesso al bucket Amazon S3](#)
- [Fase 3: Esportazione di un'istananea di MemoryDB](#)

Fase 1: creazione di un bucket Amazon S3

La procedura seguente utilizza la console Amazon S3 per creare un bucket Amazon S3 in cui esportare e archiviare lo snapshot di MemoryDB.

Come creare un bucket Amazon S3.

1. [Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.](https://console.aws.amazon.com/s3/)
2. Scegli Crea bucket.
3. In Creare bucket - Scegli un nome di bucket e una regione, esegui le seguenti operazioni:
 - a. In Nome bucket, digita un nome per il bucket Amazon S3.

- b. Dall'elenco delle regioni, scegli una AWS regione per il tuo bucket Amazon S3. Questa AWS regione deve essere la stessa AWS regione dello snapshot di MemoryDB che desideri esportare.
- c. Scegli Crea.

Per ulteriori informazioni sulla creazione di un bucket Amazon S3, consulta [Creazione di un bucket](#) nella Guida all'utente di Amazon Simple Storage Service.

Fase 2: concedere a MemoryDB l'accesso al bucket Amazon S3

AWS Le regioni introdotte prima del 20 marzo 2019 sono abilitate per impostazione predefinita. Puoi iniziare a lavorare in queste AWS regioni immediatamente. Le regioni introdotte dopo il 20 marzo 2019 sono disabilitate per impostazione predefinita. È necessario abilitare o attivare queste regioni prima di poterle utilizzare, come descritto in [Gestione delle AWS regioni](#).

Concedi a MemoryDB l'accesso al tuo bucket S3 in una regione AWS

Per creare le autorizzazioni appropriate su un bucket Amazon S3 in AWS una regione, procedi nel seguente modo.

Per concedere a MemoryDB l'accesso a un bucket S3

1. [Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/](https://console.aws.amazon.com/s3/).
2. Scegli il nome del bucket Amazon S3 in cui vuoi copiare lo snapshot. Deve essere il bucket S3 che è stato creato in [Fase 1: creazione di un bucket Amazon S3](#).
3. Scegli la scheda Autorizzazioni e in Autorizzazioni, scegli Bucket policy.
4. Aggiorna la politica per concedere a MemoryDB le autorizzazioni necessarie per eseguire operazioni:
 - Aggiungere ["Service" : "*region-full-name*.memorydb-snapshot.amazonaws.com"] a Principal.
 - Aggiungere le seguenti autorizzazioni necessarie per esportare uno snapshot nel bucket Amazon S3.
 - "s3:PutObject"
 - "s3:GetObject"
 - "s3:ListBucket"

- "s3:GetBucketAcl"
- "s3:ListMultipartUploadParts"
- "s3:ListBucketMultipartUploads"

Di seguito è riportato un esempio di come potrebbe essere la policy aggiornata.

```
{
  "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/*"
      ]
    }
  ]
}
```

Fase 3: Esportazione di un'istantanea di MemoryDB

Ora hai creato il tuo bucket S3 e concesso a MemoryDB le autorizzazioni per accedervi. Cambia la proprietà degli oggetti S3 in ACL abilitati (preferibilmente proprietario del bucket). Successivamente, puoi utilizzare la console MemoryDB, la AWS CLI o l'API MemoryDB per esportare la tua istantanea su di essa. Di seguito, si presuppone che le seguenti autorizzazioni IAM specifiche di S3 siano disponibili.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketLocation",
      "s3:ListAllMyBuckets",
      "s3:PutObject",
      "s3:GetObject",
      "s3:DeleteObject",
      "s3:ListBucket"
    ],
    "Resource": "arn:aws:s3:::*"
  }]
}
```

Esportazione di un'istantanea di MemoryDB (Console)

Il processo seguente utilizza la console MemoryDB per esportare uno snapshot in un bucket Amazon S3 in modo da potervi accedere dall'esterno di MemoryDB. Il bucket Amazon S3 deve trovarsi nella stessa AWS regione dello snapshot MemoryDB.

Per esportare uno snapshot di MemoryDB in un bucket Amazon S3

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Per visualizzare un elenco delle tue istantanee, dal riquadro di navigazione a sinistra scegli Istantanee.
3. Dall'elenco delle istantanee, scegli il pulsante di opzione a sinistra del nome dell'istantanea che desideri esportare.
4. Scegli Copia.
5. In Creare una copia del backup?, procedere come segue:
 - a. Nella casella Nome nuova istantanea, digitate un nome per la nuova istantanea.

Il nome deve essere compreso tra 1 e 1000 caratteri e dotato di codifica UTF-8.

MemoryDB aggiunge un identificatore di shard e `.rdb` al valore che inserisci qui. Ad esempio, se si immettemy-exported-snapshot, MemoryDB crea `my-exported-snapshot-0001.rdb`

- b. Dall'elenco Target S3 Location, scegli il nome del bucket Amazon S3 in cui vuoi copiare lo snapshot (il bucket in cui hai creato). [Fase 1: creazione di un bucket Amazon S3](#)

La posizione S3 di destinazione deve essere un bucket Amazon S3 nella regione dello snapshot con le seguenti autorizzazioni affinché il processo AWS di esportazione abbia successo.

- Accesso agli oggetti : Lettura e Scrittura.
- Accesso alle autorizzazioni : Lettura.

Per ulteriori informazioni, consulta [Fase 2: concedere a MemoryDB l'accesso al bucket Amazon S3](#).

- c. Scegli Copia.

Note

Se il tuo bucket S3 non dispone delle autorizzazioni necessarie a MemoryDB per esportare uno snapshot al suo interno, ricevi uno dei seguenti messaggi di errore. Torna a per aggiungere le autorizzazioni specificate e riprova [Fase 2: concedere a MemoryDB l'accesso al bucket Amazon S3](#) a esportare la tua istantanea.

- A MemoryDB non sono state concesse le autorizzazioni di LETTURA %s sul bucket S3.

Soluzione: aggiungere autorizzazioni di lettura sul bucket.

- A MemoryDB non sono state concesse le autorizzazioni di SCRITTURA %s sul bucket S3.

Soluzione: aggiungere autorizzazioni di scrittura sul bucket.

- A MemoryDB non sono state concesse le autorizzazioni READ_ACP %s sul bucket S3.

Soluzione: aggiungere Read (Lettura) per Accesso alle autorizzazioni sul bucket.

Se desideri copiare lo snapshot in un'altra AWS regione, usa Amazon S3 per copiarlo. Per ulteriori informazioni, [consulta Copiare oggetti](#) nella Guida per l'utente di Amazon Simple Storage Service.

Esportazione di un'istantanea di MemoryDB (CLI)AWS

Esporta lo snapshot in un bucket Amazon S3 utilizzando l'operazione `copy-snapshot` CLI con i seguenti parametri:

Parametri

- `--source-snapshot-name`— Nome dello snapshot da copiare.
- `--target-snapshot-name`— Nome della copia dell'istantanea.

Il nome deve essere compreso tra 1 e 1000 caratteri e dotato di codifica UTF-8.

MemoryDB aggiunge un identificatore di frammento e `.rdb` al valore immesso qui. Ad esempio, se si immette `my-exported-snapshot`, MemoryDB crea `my-exported-snapshot-0001.rdb`.

- `--target-bucket`— Nome del bucket Amazon S3 in cui desideri esportare lo snapshot. Una copia dello snapshot viene creata nel bucket specificato.

`--target-bucket` Affinché il processo di esportazione abbia successo, deve essere un bucket Amazon S3 AWS nella regione dello snapshot con le seguenti autorizzazioni.

- Accesso agli oggetti : Lettura e Scrittura.
- Accesso alle autorizzazioni : Lettura.

Per ulteriori informazioni, consulta [Fase 2: concedere a MemoryDB l'accesso al bucket Amazon S3](#).

La seguente operazione copia uno snapshot in `my-s3-bucket`.

Per 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
```

Per 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

Se il bucket S3 non dispone delle autorizzazioni necessarie a MemoryDB per esportare un'istantanea al suo interno, viene visualizzato uno dei seguenti messaggi di errore. Torna a per aggiungere le autorizzazioni specificate e riprova [Fase 2: concedere a MemoryDB l'accesso al bucket Amazon S3](#) a esportare la tua istantanea.

- A MemoryDB non sono state concesse le autorizzazioni di LETTURA %s sul bucket S3.
Soluzione: aggiungere autorizzazioni di lettura sul bucket.
- A MemoryDB non sono state concesse le autorizzazioni di SCRITTURA %s sul bucket S3.
Soluzione: aggiungere autorizzazioni di scrittura sul bucket.
- A MemoryDB non sono state concesse le autorizzazioni READ_ACP %s sul bucket S3.
Soluzione: aggiungere Read (Lettura) per Accesso alle autorizzazioni sul bucket.

Per ulteriori informazioni, consulta la sezione `copy-snapshot` nella Documentazione di riferimento della AWS CLI .

Se desideri copiare lo snapshot in un'altra AWS regione, usa Amazon S3 copy. Per ulteriori informazioni, [consulta Copiare oggetti](#) nella Guida per l'utente di Amazon Simple Storage Service.

Esportazione di uno snapshot di MemoryDB (API MemoryDB)

Esporta lo snapshot in un bucket Amazon S3 utilizzando CopySnapshot l'operazione API con questi parametri.

Parametri

- `SourceSnapshotName`— Nome dello snapshot da copiare.
- `TargetSnapshotName`— Nome della copia dell'istantanea.

Il nome deve essere compreso tra 1 e 1000 caratteri e dotato di codifica UTF-8.

MemoryDB aggiunge un identificatore di shard e `.rdb` al valore che inserisci qui. Ad esempio, se inserisci `my-exported-snapshot`, ottieni `my-exported-snapshot-0001.rdb`.

- **TargetBucket**— Nome del bucket Amazon S3 in cui desideri esportare lo snapshot. Una copia dello snapshot viene creata nel bucket specificato.

TargetBucket Affinché il processo di esportazione abbia successo, deve essere un bucket Amazon S3 AWS nella regione dello snapshot con le seguenti autorizzazioni.

- Accesso agli oggetti : Lettura e Scrittura.
- Accesso alle autorizzazioni : Lettura.

Per ulteriori informazioni, consulta [Fase 2: concedere a MemoryDB l'accesso al bucket Amazon S3](#).

L'esempio seguente crea una copia di uno snapshot automatico nel bucket Amazon my-s3-bucket 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

Se il tuo bucket S3 non dispone delle autorizzazioni necessarie a MemoryDB per esportare uno snapshot al suo interno, ricevi uno dei seguenti messaggi di errore. Torna a per aggiungere le autorizzazioni specificate e riprova [Fase 2: concedere a MemoryDB l'accesso al bucket Amazon S3](#) a esportare la tua istantanea.

- A MemoryDB non sono state concesse le autorizzazioni di LETTURA %s sul bucket S3.

Soluzione: aggiungere autorizzazioni di lettura sul bucket.

- A MemoryDB non sono state concesse le autorizzazioni di SCRITTURA %s sul bucket S3.

Soluzione: aggiungere autorizzazioni di scrittura sul bucket.

- A MemoryDB non sono state concesse le autorizzazioni READ_ACP %s sul bucket S3.

Soluzione: aggiungere Read (Lettura) per Accesso alle autorizzazioni sul bucket.

Per ulteriori [CopySnapshot](#) informazioni, consulta.

Se desideri copiare lo snapshot in un'altra AWS regione, usa Amazon S3 copy per copiare lo snapshot esportato nel bucket Amazon S3 in un'altra regione. AWS Per ulteriori informazioni, [consulta Copiare oggetti](#) nella Guida per l'utente di Amazon Simple Storage Service.

Ripristino da uno snapshot

Puoi ripristinare i dati da un file snapshot MemoryDB o ElastiCache Redis .rdb in un nuovo cluster in qualsiasi momento.

Il processo di ripristino di MemoryDB per Redis supporta quanto segue:

- Migrazione da uno o più file snapshot .rdb creati per Redis a un cluster MemoryDB. ElastiCache
- I file .rdb devono essere inseriti in S3 per eseguire il ripristino.
- Specificando un numero di shard nel nuovo cluster diverso dal numero di shard nel cluster utilizzato per creare il file snapshot.
 - Specifica di un tipo di nodo diverso per il nuovo cluster, ovvero più grande o più piccolo. Se esegui il dimensionamento a un tipo di nodo più piccolo, assicurati che il nuovo tipo di nodo disponga di memoria sufficiente per i dati e l'overhead Redis.
 - Configurazione degli slot del nuovo cluster MemoryDB in modo diverso rispetto al cluster utilizzato per creare il file di istantanea.

Important

- I cluster MemoryDB non supportano più database. Pertanto, quando si esegue il ripristino su MemoryDB, il ripristino non riesce se il file.rdb fa riferimento a più di un database.
- Non è possibile ripristinare un'istantanea da un cluster che utilizza il tiering dei dati (ad esempio, il tipo di nodo r6gd) in un cluster che non utilizza il tiering dei dati (ad esempio, il tipo di nodo r6g).

La possibilità di apportare modifiche durante il ripristino di un cluster da un'istantanea dipende dalle scelte effettuate. Queste scelte vengono effettuate nella pagina Restore Cluster quando si utilizza la console MemoryDB per il ripristino. È possibile effettuare queste scelte impostando i valori dei parametri quando si utilizza l'API AWS CLI o MemoryDB per il ripristino.

Durante l'operazione di ripristino, MemoryDB crea il nuovo cluster e quindi lo popola con i dati del file snapshot. Una volta completato questo processo, il cluster viene riscaldato e pronto ad accettare le richieste.

⚠ Important

Prima di procedere, assicurati di aver creato un'istantanea del cluster da cui desideri eseguire il ripristino. Per ulteriori informazioni, consulta [Creazione di istantanee manuali](#).

Se desideri eseguire il ripristino da un'istantanea creata esternamente, consulta. [Seminare un nuovo cluster con un'istantanea creata esternamente](#)

Le seguenti procedure mostrano come ripristinare un'istantanea in un nuovo cluster utilizzando la console MemoryDB, l'API MemoryDB o l'API MemoryDB. AWS CLI

Ripristino da un'istantanea (console)

Per ripristinare un'istantanea in un nuovo cluster (console)

1. [Accedere AWS Management Console e aprire la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. Nel pannello di navigazione, scegli Istantanee.
3. Nell'elenco delle istantanee, scegli il pulsante accanto al nome dell'istantanea da cui desideri eseguire il ripristino.
4. Scegli Azioni, quindi scegli Ripristina
5. In Configurazione del cluster, inserisci quanto segue:
 - a. Nome del cluster: obbligatorio. Il nome del nuovo cluster.
 - b. Descrizione: facoltativa. La descrizione del nuovo cluster.
6. Completa la sezione Gruppi di sottoreti:
 - Per i gruppi di sottoreti, crea un nuovo gruppo di sottoreti o scegline uno esistente dall'elenco disponibile che desideri applicare a questo cluster. Se ne stai creando uno nuovo:
 - Inserisci un nome
 - Inserisci una descrizione
 - Se è stata attivata la funzione Multi-AZ, il gruppo di sottoreti deve contenere almeno due sottoreti che risiedono in zone di disponibilità diverse. Per ulteriori informazioni, consulta [Sottoreti e gruppi di sottoreti](#).

- Se stai creando un nuovo gruppo di sottoreti e non disponi di un VPC esistente, ti verrà chiesto di creare un VPC. Per ulteriori informazioni, consultare [Che cos'è Amazon VPC?](#) nella Guida per l'utente di Amazon VPC

7. Completa la sezione Impostazioni del cluster:

- a. Per la compatibilità della versione Redis, accetta l'impostazione predefinita `6.0`.
- b. Per Port, accetta la porta Redis predefinita `6379` o, se hai un motivo per utilizzare una porta diversa, inserisci il numero di porta..
- c. Per Parameter group, accettate il gruppo di `default.memorydb-redis6` parametri.

I gruppi di parametri controllano i parametri di runtime del cluster. Per ulteriori informazioni sui gruppi di parametri, consulta [Parametri specifici di Redis](#).

- d. Per Tipo di nodo, scegliete un valore per il tipo di nodo (insieme alla dimensione della memoria associata) che desiderate.

Se scegli un membro della famiglia di tipi di nodi `r6gd`, abiliterai automaticamente il tiering dei dati nel cluster. Per ulteriori informazioni, consulta [Tiering di dati](#).

- e. Per Numero di shard, scegli il numero di shard che desideri per questo cluster.

È possibile modificare il numero di shard nel cluster in modo dinamico. Per ulteriori informazioni, consulta [Dimensionamento di cluster di MemoryDB](#).

- f. In Replicas per shard (Repliche per partizione): scegliere il numero di nodi di replica di lettura per ogni partizione.

Esistono le seguenti restrizioni;

- Se hai abilitato la funzione Multi-AZ, assicurati di avere almeno una replica per ogni partizione.
- Quando utilizzi la console per creare il cluster, il numero delle repliche è lo stesso per ogni partizione.


- g. Seleziona Next (Successivo).

- h. Completa la sezione Impostazioni avanzate:

- i. In Security groups (Gruppi di sicurezza), scegliere i gruppi di sicurezza per il cluster. Un gruppo di sicurezza si comporta come un firewall, controllando l'accesso di rete al cluster. È possibile utilizzare il gruppo di sicurezza di default per il VPC o crearne uno nuovo.

Per ulteriori informazioni sui gruppi di sicurezza, consulta [Gruppi di sicurezza per il VPC](#) nella Guida per l'utente di Amazon VPC.

- ii. I dati vengono crittografati nei seguenti modi:
 - Crittografia dei dati inattivi : Consente la crittografia dei dati memorizzati su disco. Per ulteriori informazioni, consultare [Crittografia dei dati inattivi](#).

 Note

È possibile fornire una chiave di crittografia diversa scegliendo la chiave AWS KMS gestita dal cliente e scegliendo la chiave.

- Crittografia dei dati in transito : Consente la crittografia dei dati in trasferimento. Questo è abilitato per impostazione predefinita. Per maggiori informazioni, consultare [Crittografia dei dati in transito](#).

Se non si seleziona alcuna crittografia, verrà creata una lista di controllo degli accessi aperta denominata «accesso aperto» con un utente predefinito. Per ulteriori informazioni, consulta [Autenticazione degli utenti con elenchi di controllo degli accessi \(ACL\)](#).

- iii. Per Snapshot, specificate facoltativamente un periodo di conservazione delle istantanee e una finestra per le istantanee. Per impostazione predefinita, è selezionata l'opzione Abilita istantanee automatiche.
- iv. Per la finestra Manutenzione, specificare facoltativamente una finestra di manutenzione. La finestra di manutenzione è l'orario, generalmente di un'ora, ogni settimana in cui MemoryDB pianifica la manutenzione del sistema per il cluster. È possibile consentire a MemoryDB di scegliere il giorno e l'ora per la finestra di manutenzione (nessuna preferenza), oppure è possibile scegliere autonomamente il giorno, l'ora e la durata (Specificare la finestra di manutenzione). Se dagli elenchi si sceglie Specify maintenance window (Specifica finestra di manutenzione), selezionare Start day (Giorno di inizio), Start time (Ora di inizio) e Duration (Durata) (in ore) per la finestra di manutenzione. Tutti gli orari si intendono in formato UCT.


Per ulteriori informazioni, consulta [Gestione della manutenzione](#).

- v. In Notifications (Notifiche), scegliere un argomento esistente di Amazon Simple Notification Service (Amazon SNS) o scegliere l'input manuale dell'ARN nell'Amazon

Resource Name (ARN) dell'argomento. Amazon SNS permette di inviare notifiche ai dispositivi intelligenti connessi a Internet. Le notifiche sono disabilitate per impostazione predefinita. [Per ulteriori informazioni, consulta https://aws.amazon.com/sns/](https://aws.amazon.com/sns/).

- i. Per i tag, puoi facoltativamente applicare tag per cercare e filtrare i cluster o tenere traccia AWS dei costi.
- j. Riesaminare le voci e le selezioni, quindi apportare le eventuali correzioni. Al termine, scegliere Create cluster (Crea cluster) per avviare il cluster o Cancel (Annulla) per annullare l'operazione.

Quando lo stato del cluster diventa disponibile, è possibile concedere a EC2 l'accesso al cluster, connettersi al cluster e iniziare a utilizzarlo. Per ulteriori informazioni, consulta [Passaggio 2: autorizzare l'accesso al cluster](#) e [Fase 3: Connect al cluster](#).

 Important

Non appena il cluster diventa disponibile, viene addebitata ogni ora o frazione di ora in cui il cluster è attivo, anche se non viene effettivamente utilizzato. Per evitare di sostenere i costi del cluster, è necessario eliminarlo. Per informazioni, consulta [Fase 4: Eliminazione di un cluster](#).

Ripristino da un'istantanea (CLI AWS)

Quando utilizzate una delle due `create-cluster` operazioni, assicuratevi di includere il parametro `--snapshot-name` o di `--snapshot-arns` inserire nel nuovo cluster i dati dell'istantanea.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Creazione di un cluster \(AWS CLI\)](#) nella Guida per l'utente di MemoryDB.
- [create-cluster](#) nel Command Reference. AWS CLI

Ripristino da un'istantanea (API MemoryDB)

È possibile ripristinare un'istantanea di MemoryDB utilizzando l'operazione API MemoryDB.

CreateCluster

Quando utilizzate l'`CreateCluster` operazione, assicuratevi di includere il parametro `SnapshotName` o di `SnapshotArns` inserire nel nuovo cluster i dati dell'istantanea.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Creazione di un cluster \(API MemoryDB\)](#) nella Guida per l'utente di MemoryDB.
- [CreateCluster](#) nel riferimento alle API di MemoryDB.

Seminare un nuovo cluster con un'istantanea creata esternamente

Quando si crea un nuovo cluster MemoryDB, è possibile effettuare il seeding con i dati di un file snapshot Redis .rdb.

Per effettuare il seeding di un nuovo cluster MemoryDB da uno snapshot di MemoryDB o per uno snapshot Redis, vedi. ElastiCache [Ripristino da uno snapshot](#)

Quando usi un file Redis .rdb per seminare un nuovo cluster MemoryDB, puoi fare quanto segue:

- Specificate un numero di shard nel nuovo cluster. Questo numero può essere diverso dal numero di shard nel cluster utilizzato per creare il file snapshot.
- Specificate un tipo di nodo diverso per il nuovo cluster, più grande o più piccolo di quello utilizzato nel cluster che ha creato l'istantanea. Se esegui il dimensionamento a un tipo di nodo più piccolo, assicurati che il nuovo tipo di nodo disponga di memoria sufficiente per i dati e l'overhead Redis.

Important

- È necessario assicurarsi che i dati dell'istantanea non superino le risorse del nodo.

Se l'istantanea è troppo grande, il cluster risultante ha uno stato di `restore-failed` In tal caso, occorre eliminare il cluster e ricominciare.

Per un elenco completo dei tipi e delle specifiche dei nodi, vedere [Parametri specifici del tipo di nodo di MemoryDB](#).

- È possibile eseguire la crittografia di un file .rdb Redis con Amazon S3 solo con crittografia lato server (SSE-S3). Per ulteriori informazioni, consulta [Protezione dei dati con la crittografia lato server](#).

Fase 1: Creare un'istantanea redis su un cluster esterno

Per creare l'istantanea per il seeding del cluster MemoryDB

1. Connettersi all'istanza Redis esistente.
2. Esegui Redis BGSAVE o l'SAVEoperazione per creare un'istantanea. Prendere nota della posizione del file .rdb.

BGSAVE è asincrona e non blocca altri client durante l'elaborazione. Per ulteriori informazioni, consulta [BGSAVE](#) presso il sito Web Redis.

SAVE è sincrona e blocca altri processi finché non è terminata. Per ulteriori informazioni, consulta [SAVE](#) presso il sito Web Redis.

Per ulteriori informazioni sulla creazione di un'istantanea, consulta la [persistenza di Redis sul sito Web di Redis](#).

Fase 2: creazione di un bucket Amazon S3 e una cartella

Dopo aver creato il file snapshot, devi caricarlo in una cartella all'interno di un bucket Amazon S3. A questo scopo, tale bucket deve contenere un bucket Amazon S3 e una cartella. Se disponi già di un bucket Amazon S3 e una cartella con le autorizzazioni appropriate, puoi passare a [Passaggio 3: carica la tua istantanea su Amazon S3](#).

Come creare un bucket Amazon S3.

1. [Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/](https://console.aws.amazon.com/s3/).
2. Seguire le istruzioni per creare un bucket Amazon S3 in [Creazione di un bucket](#) nella Guida per l'utente di Amazon Simple Storage Service.

Il nome del bucket Amazon S3 deve essere conforme a DNS. Altrimenti, MemoryDB non potrà accedere al tuo file di backup. Le regole per la conformità DNS sono:

- I nomi devono avere una lunghezza compresa fra 3 e 63 caratteri.
- I nomi devono contenere una serie di una o più etichette separate da un punto (.) in cui ciascuna etichetta:
 - Inizia con una lettera minuscola o un numero.
 - Finisce con una lettera minuscola o un numero.
 - Contiene solo lettere minuscole, numeri e trattini.
- Non deve avere il formato di un indirizzo IP (ad esempio, 192.0.2.0).

Ti consigliamo vivamente di creare il tuo bucket Amazon S3 nella stessa AWS regione del tuo nuovo cluster MemoryDB. Questo approccio garantisce la massima velocità di trasferimento dei dati quando MemoryDB legge il file.rdb da Amazon S3.

Note

Per tenere i dati al sicuro, crea autorizzazioni al bucket Amazon S3 il più possibile restrittive. Allo stesso tempo, le autorizzazioni devono comunque consentire l'utilizzo del bucket e del suo contenuto per il seeding del nuovo cluster MemoryDB.

Per aggiungere una cartella a un bucket Amazon S3

1. [Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.](https://console.aws.amazon.com/s3/)
2. Scegliere il nome del bucket in cui caricare il file .rdb.
3. Scegliere Create folder (Crea cartella).
4. Immettere un nome per la nuova cartella.
5. Selezionare Salva.

Prendi nota del nome del bucket e del nome della cartella.

Passaggio 3: carica la tua istantanea su Amazon S3

Ora, carica il file .rdb creato in [Fase 1: Creare un'istantanea redis su un cluster esterno](#). Caricalo nel bucket Amazon S3 e nella cartella creata in [Fase 2: creazione di un bucket Amazon S3 e una cartella](#). Per ulteriori informazioni su questa attività, consulta [Caricamento](#) di oggetti. Tra le fasi 2 e 3, scegliere il nome della cartella creata.

Per caricare il file .rdb in una cartella Amazon S3

1. [Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.](https://console.aws.amazon.com/s3/)
2. Scegliere il nome del bucket Amazon S3 creato nella fase 2.
3. Scegliere il nome della cartella creata nella Fase 2.

4. Scegli Carica.
5. Scegliere Add files (Aggiungi file).
6. Individuare il file o i file da caricare, quindi scegliere il file o i file. Per scegliere più file, tenere premuto il tasto CTRL durante la selezione di ciascun nome file.
7. Seleziona Apri.
8. Verifica che il file o i file corretti siano elencati nella pagina di caricamento, quindi scegli Carica.

Nota il percorso nel file .rdb. Ad esempio, se il nome del bucket è myBucket e il percorso è myFolder/redis.rdb, digitare myBucket/myFolder/redis.rdb. Questo percorso è necessario per eseguire il seeding del nuovo cluster con i dati contenuti in questa istantanea.

Per ulteriori informazioni, consulta le [regole di denominazione dei bucket nella Guida](#) per l'utente di Amazon Simple Storage Service.

Passaggio 4: concedere a MemoryDB l'accesso in lettura al file.rdb

AWS Le regioni introdotte prima del 20 marzo 2019 sono abilitate per impostazione predefinita. Puoi iniziare a lavorare in queste AWS regioni immediatamente. Le regioni introdotte dopo il 20 marzo 2019 sono disabilitate per impostazione predefinita. È necessario abilitare o attivare queste regioni prima di poterle utilizzare, come descritto in [Gestione delle AWS regioni](#).

Concedi a MemoryDB l'accesso in lettura al file.rdb

Per concedere a MemoryDB l'accesso in lettura al file snapshot

1. [Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/](https://console.aws.amazon.com/s3/).
2. Scegliere il nome del bucket S3 contenente il file .rdb.
3. Scegliere il nome della cartella contenente il file .rdb.
4. Scegli il nome del tuo file snapshot .rdb. Il nome del file selezionato viene visualizzato sopra le schede nella parte superiore della pagina.
5. Scegli la scheda Autorizzazioni.
6. Sotto Autorizzazioni, scegli Policy bucket e seleziona Modifica.
7. Aggiorna la politica per concedere a MemoryDB le autorizzazioni necessarie per eseguire le operazioni:

- Aggiungere ["Service" : "*region-full-name*.memorydb-snapshot.amazonaws.com"] a Principal.
- Aggiungere le seguenti autorizzazioni necessarie per l'esportazione di uno snapshot nel bucket Amazon S3:
 - "s3:GetObject"
 - "s3:ListBucket"
 - "s3:GetBucketAcl"

Di seguito è riportato un esempio di come potrebbe essere la policy aggiornata.

```
{
  "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. Selezionare Salva.

Fase 5: Esegui il seeding del cluster MemoryDB con i dati del file.rdb

Ora siete pronti per creare un cluster MemoryDB e inserirlo con i dati del file.rdb. Per creare il cluster, segui le istruzioni in [Creazione di un cluster MemoryDB](#)

Il metodo utilizzato per indicare a MemoryDB dove trovare lo snapshot Redis caricato su Amazon S3 dipende dal metodo utilizzato per creare il cluster:

Semina il cluster MemoryDB con i dati del file.rdb

- Utilizzo della console MemoryDB

Dopo aver scelto il motore Redis, espandi la sezione Advanced Redis settings (Impostazioni avanzate di Redis) e individua Import data to cluster (Importazione dei dati in un cluster). Nella casella Seed RDB file S3 location (Inizializza posizione Amazon S3 del file RDB), digita il percorso per i file. Se disponi di più file .rdb, digita il percorso per ogni file in un elenco separato da virgole. Il percorso Amazon S3 appare simile a *myBucket/myFolder/myBackupFilename*.rdb.

- Usando il AWS CLI

Se utilizzi l'operazione `create-cluster` o `create-cluster`, utilizza il parametro `--snapshot-arns` per specificare un ARN completo per ogni file .rdb. Ad esempio, `arn:aws:s3:::myBucket/myFolder/myBackupFilename`.rdb. L'ARN deve essere risolto nei file di snapshot archiviati in Amazon S3.

- Utilizzando l'API MemoryDB

Se si utilizza l'operazione API `CreateCluster` MemoryDB `CreateCluster` o l'operazione MemoryDB, utilizzare il parametro `SnapshotArns` per specificare un ARN completo per ogni file.rdb. Ad esempio, `arn:aws:s3:::myBucket/myFolder/myBackupFilename`.rdb. L'ARN deve essere risolto nei file di snapshot archiviati in Amazon S3.

Durante il processo di creazione del cluster, i dati dello snapshot vengono scritti nel cluster. È possibile monitorare l'avanzamento visualizzando i messaggi degli eventi MemoryDB. Per fare ciò, consulta la console MemoryDB e scegli Eventi. È inoltre possibile utilizzare l'interfaccia a riga di comando di AWS MemoryDB o l'API MemoryDB per ottenere messaggi di eventi.

Taggare le istantanee

È possibile assegnare i propri metadati a ciascuna istantanea sotto forma di tag. I tag consentono di classificare le istantanee in diversi modi, ad esempio per scopo, proprietario o ambiente. Questa caratteristica è molto utile quando hai tante risorse dello stesso tipo in quanto puoi rapidamente individuare una risorsa specifica in base ai tag assegnati. Per ulteriori informazioni, consulta [Risorse cui è possibile associare tag](#).

I tag di allocazione dei costi sono un mezzo per tenere traccia dei costi di più AWS servizi raggruppando le spese sulle fatture in base ai valori dei tag. Per ulteriori informazioni sui tag di allocazione dei costi, consulta la sezione relativa all'[uso dei tag di allocazione dei costi](#).

Utilizzando la console MemoryDB AWS CLI, l'API MemoryDB o l'API MemoryDB è possibile aggiungere, elencare, modificare, rimuovere o copiare i tag di allocazione dei costi nelle istantanee. Per ulteriori informazioni, consulta [Monitoraggio dei costi con i tag di allocazione dei costi](#).

Eliminazione di uno snapshot

Un'istantanea automatica viene eliminata automaticamente alla scadenza del relativo limite di conservazione. Se si elimina un cluster, vengono eliminate anche tutte le relative istantanee automatiche.

MemoryDB fornisce un'operazione API di eliminazione che consente di eliminare un'istantanea in qualsiasi momento, indipendentemente dal fatto che l'istantanea sia stata creata automaticamente o manualmente. Poiché le istantanee manuali non hanno un limite di conservazione, l'eliminazione manuale è l'unico modo per rimuoverle.

È possibile eliminare un'istantanea utilizzando la console MemoryDB AWS CLI, o l'API MemoryDB.

Eliminazione di un'istantanea (Console)

La procedura seguente elimina un'istantanea utilizzando la console MemoryDB.

Per eliminare uno snapshot

1. [Accedere AWS Management Console e aprire la console MemoryDB for Redis all'indirizzo `https://console.aws.amazon.com/memorydb/`.](https://console.aws.amazon.com/memorydb/)
2. Nel riquadro di navigazione a sinistra, scegli Istantanee.
Viene visualizzata la schermata Istantanee con un elenco delle istantanee.
3. Scegli il pulsante radio a sinistra del nome dell'istantanea che desideri eliminare.
4. Scegliere Actions (Operazioni), quindi selezionare Delete (Elimina VPC).
5. Se desideri eliminare questa istantanea, inseriscila **delete** nella casella di testo e scegli Elimina. Per annullare l'eliminazione, scegli Annulla. Lo stato cambia in eliminazione.

Eliminazione di un'istantanea (CLI AWS)

Utilizzate l' AWS CLI operazione delete-snapshot con il seguente parametro per eliminare un'istantanea.

- `--snapshot-name`— Nome dell'istantanea da eliminare.

Il codice seguente elimina l'istantanea. `myBackup`

```
aws memorydb delete-snapshot --snapshot-name myBackup
```

Per ulteriori informazioni, vedere [delete-snapshot](#) in AWS CLI guida di riferimento del comando.

Eliminazione di un'istantanea (API MemoryDB)

Utilizzate l'operazione DeleteSnapshot API con il seguente parametro per eliminare un'istantanea.

- SnapshotName— Nome dell'istantanea da eliminare.

Il codice seguente elimina l'istantanea. 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>
```

Per ulteriori informazioni, vedere. [DeleteSnapshot](#)

Dimensionamento

Raramente la quantità dei dati che un'applicazione deve elaborare è statica. Aumenta e diminuisce secondo le fluttuazioni della domanda del tuo business. Se gestisci autonomamente le tue applicazioni, devi fornire hardware sufficiente per i picchi di domanda, il che può essere costoso. Utilizzando MemoryDB for Redis puoi dimensionare per soddisfare la domanda corrente, pagando solo per ciò che usi.

I contenuti seguenti ti danno la possibilità di trovare l'argomento adatto alle operazioni di scalabilità di tuo interesse.

Scalabilità della memoria DB

Operazione	Memoria DB
Dimensionamento orizzontale	Reshard online e ribilanciamento dello shard per MemoryDB

Operazione	Memoria DB	
Modifica dei tipi di nodo	Ridimensionamento verticale online tramite la modifica del tipo di nodo	
Modifica del numero di partizioni	Dimensionamento di cluster di MemoryDB	

Dimensionamento di cluster di MemoryDB

Al variare della domanda di cluster, è possibile decidere di migliorare le prestazioni o ridurre i costi modificando il numero di shard nel tuo cluster MemoryDB. Per questa operazione si consiglia di utilizzare il dimensionamento orizzontale online, poiché consente ai cluster di continuare a servire le richieste durante il processo di dimensionamento.

È possibile decidere di ridimensionare il cluster in presenza delle seguenti condizioni:

- Utilizzo elevato di memoria:

Se i nodi nel cluster sono sottoposti a utilizzo elevato di memoria, è possibile decidere di aumentare le dimensioni per disporre delle risorse necessarie per migliorare l'archiviazione dei dati e servire le richieste.

È possibile determinare se i nodi sono sotto pressione di memoria monitorando le seguenti metriche: `FreeableMemorySwapUsage`, e `BytesUsedForMemoryDB`.

- Collo di bottiglia della CPU o della rete:

Se si riscontrano problemi di latenza/throughput del cluster, è possibile aumentare le dimensioni per risolvere tali problemi.

È possibile monitorare i livelli di latenza e velocità effettiva monitorando le seguenti metriche: utilizzo della CPU `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections`, e `NewConnections`.

- Il cluster è sovradimensionato:

La domanda corrente sul cluster è tale che la riduzione delle dimensioni non compromette le prestazioni e riduce i costi.

Puoi monitorare l'utilizzo del tuo cluster per determinare se puoi scalare in sicurezza o meno utilizzando le seguenti metriche: `FreeableMemory`, `SwapUsage`, `BytesUsedForMemoryDB`, `CPUUtilization` `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections`, e `NewConnections`.

Impatto del dimensionamento sulle prestazioni

Quando si effettua il dimensionamento utilizzando il processo offline, il cluster è offline per una porzione significativa del processo, pertanto non è disponibile per servire le richieste. Quando si effettua il dimensionamento utilizzando il metodo online, poiché il dimensionamento è un'operazione di calcolo intensiva, si registra un peggioramento delle prestazioni ma, nonostante tutto, il cluster

continua a servire richieste mediante l'operazione di scalabilità. Il livello di peggioramento riscontrato dipende dall'utilizzo normale della CPU e dai dati.

Esistono due modi per dimensionare il cluster MemoryDb: ridimensionamento orizzontale e verticale.

- Il ridimensionamento orizzontale consente di modificare il numero di frammenti nel cluster aggiungendo o rimuovendo frammenti. Il processo di resharding online consente il ridimensionamento in entrambe le direzioni mentre il cluster continua a servire le richieste in arrivo.
- Ridimensionamento verticale: ridimensiona il cluster tramite la modifica del tipo di nodo. Il processo di ridimensionamento verticale online consente il ridimensionamento in entrambe le direzioni mentre il cluster continua a servire le richieste in arrivo.

Se stai riducendo le dimensioni e la capacità di memoria del cluster, tramite ridimensionamento orizzontale o dimensionamento verso il basso, assicurati che la nuova configurazione disponga di memoria sufficiente per i dati e l'overhead di Redis.

Reshard offline e ribilanciamento dello shard per MemoryDB

Il vantaggio principale che ottieni dalla riconfigurazione offline degli shard è che puoi fare molto di più che semplicemente aggiungere o rimuovere frammenti dal tuo cluster. Quando esegui il resharding offline, oltre a modificare il numero di partizioni nel cluster, puoi fare quanto segue:

- Cambia il tipo di nodo del tuo cluster.
- Effettuare l'upgrade a una versione del motore più recente.

Note

Il resharding offline non è supportato nei cluster con il tiering di dati abilitato. Per ulteriori informazioni, consulta [Tiering di dati](#).

Lo svantaggio principale della riconfigurazione shard offline è che il cluster è offline a partire dalla fase di ripristino del processo e continua a essere offline fino agli aggiornamenti degli endpoint nell'applicazione. Il periodo di tempo in cui il cluster rimane offline dipende dalla quantità di dati nel cluster.

Per riconfigurare il cluster Shards MemoryDB offline

1. Creare una snapshot manuale del cluster MemoryDB esistente. Per ulteriori informazioni, consulta [Creazione di istantanee manuali](#).
2. Crea un nuovo cluster eseguendo il ripristino dall'istantanea. Per ulteriori informazioni, consulta [Ripristino da uno snapshot](#).
3. Aggiornare gli endpoint nell'applicazione agli endpoint del nuovo cluster. Per ulteriori informazioni, consulta [Individuazione degli endpoint di connessione](#).

Reshard online e ribilanciamento dello shard per MemoryDB

Utilizzando il resharding e ribilanciamento partizioni online con MemoryDB, puoi dimensionare il tuo MemoryDB in modo dinamico senza tempi di inattività. Questo approccio indica che il cluster può continuare a servire le richieste anche durante il dimensionamento o il ribilanciamento.

Puoi eseguire le operazioni indicate di seguito:

- Scale out: aumenta la capacità di lettura e scrittura aggiungendo frammenti al tuo cluster MemoryDB.

Se si aggiungono uno o più partizioni al cluster, il numero di nodi in ogni nuovo shard è lo stesso del numero di nodi nel più piccolo tra le partizioni esistenti.

- Dimensionamento verso l'alto: riduzione della capacità di lettura e scrittura, e quindi dei costi, rimuovendo le partizioni dal cluster MemoryDB.

Attualmente, il resharding online di MemoryDB prevede le seguenti limitazioni:

- Gli slot, gli spazi chiave e gli elementi grandi prevedono delle limitazioni:

Se alcune chiavi in una partizione contengono un elemento grande, quelle chiavi non possono essere migrate a un nuovo shard durante l'aumento delle dimensioni o il ribilanciamento. Questa caratteristica può produrre partizioni non bilanciati.

Se alcune chiavi in una partizione contengono un elemento grande (di dimensioni superiori a 256 MB dopo la serializzazione), quella partizione non viene eliminata se le dimensioni diminuiscono. Con questa caratteristica alcune partizioni potrebbero non essere eliminate.

- Durante il dimensionamento verso l'alto, il numero di nodi in ogni nuovo shard è uguale al numero di nodi nelle partizioni esistenti.

Per ulteriori informazioni, consulta [Best practice. Dimensionamento di cluster online](#).

Puoi dimensionare orizzontalmente o ribilanciare i cluster MemoryDB utilizzando l'AWS Management Console, l'AWS CLI, o MemoryDB.

Aggiunta delle partizioni con il resharding online

Puoi aggiungere le partizioni al tuo cluster MemoryDB utilizzando l'AWS Management Console, l'AWS CLI, o MemoryDB.

Aggiunta delle partizioni (Console)

Puoi utilizzare l'AWS Management Console per aggiungere una o più partizioni al cluster MemoryDB. Questo processo viene descritto di seguito.

1. Accedi all'AWS Management Console e apri la console MemoryDB for Redis all'[indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. Nell'elenco dei cluster, scegliere il nome del cluster da cui aggiungere una partizione.
3. Nella scheda Frammenti e nodi, scegli Aggiungi/Elimina frammenti.
4. In Nuovo numero di frammenti, inserisci il numero di frammenti che desideri.
5. Scegli Conferma per mantenere le modifiche o Annulla per annullare.

Aggiunta delle partizioni (AWS CLI)

La procedura seguente descrive come riconfigurare le partizioni nel cluster MemoryDB aggiungendo le partizioni che utilizzano l'AWS CLI.

Utilizzare i seguenti parametri con `update-cluster`.

Parametri

- `--cluster-name` – Obbligatorio. Specifica su quale cluster (cluster) deve essere eseguita l'operazione di riconfigurazione dello shard.
- `--shard-configuration` – Obbligatorio. Consente di impostare il numero di partizioni.
 - `ShardCount`: Imposta questa proprietà per specificare il numero di partizioni desiderato.

Example

L'esempio seguente modifica il numero di frammenti nel cluster `portandolomy-cluster` a 2.

Per Linux, macOS o Unix:

```
aws memorydb update-cluster \  
  --cluster-name my-cluster \  
  --shard-configuration \  
    ShardCount=2
```

Per Windows:

```
aws memorydb update-cluster ^  
  --cluster-name my-cluster ^  
  --shard-configuration ^  
    ShardCount=2
```

Restituisce la risposta JSON seguente:

```
{  
  "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  
  }  
}
```

Per visualizzare i dettagli del cluster aggiornato una volta che il suo stato cambia da aggiornamento a disponibile, usa il comando seguente:

Per Linux, macOS o Unix:

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

Per Windows:

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

Restituirà la seguente risposta 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
    }
]
}

```

Per ulteriori informazioni, consulta [update-cluster](#) nel Riferimento aiAWS CLI comandi.

Aggiunta di frammenti (API di MemoryDb)

Puoi utilizzare l'API MemoryDB per riconfigurare le partizioni nel tuo cluster MemoryDB online utilizzando l'UpdateClusteroperazione.

Utilizzare i seguenti parametri con UpdateCluster.

Parametri

- **ClusterName** – Obbligatorio. Specifica su quale cluster deve essere eseguita l'operazione di riconfigurazione dello shard.
- **ShardConfiguration** – Obbligatorio. Consente di impostare il numero di partizioni.
 - **ShardCount**: Imposta questa proprietà per specificare il numero di partizioni desiderato.

Per ulteriori informazioni, vedere [UpdateCluster](#).

Rimozione delle partizioni con il resharding online

Puoi rimuovere le partizioni dal tuo cluster MemoryDB utilizzando l'API AWS Management Console, AWS CLI, o MemoryDB.

Rimozione delle partizioni (Console)

La procedura seguente descrive come riconfigurare le partizioni nel cluster MemoryDB rimuovendo le partizioni che utilizzano la AWS Management Console.

Important

Prima di rimuovere le partizioni dal cluster, MemoryDB verifica che tutti i dati siano adatti alle partizioni rimanenti. Se i dati sono adatti, le partizioni vengono eliminate dal cluster come richiesto. Se i dati non rientrano negli shard rimanenti, il processo viene terminato e il cluster viene lasciato con la stessa configurazione di shard di prima della richiesta.

Puoi utilizzare la AWS Management Console per rimuovere una o più partizioni dal cluster MemoryDB. Non puoi rimuovere tutti le partizioni in un cluster. Dovrai invece eliminare il cluster. Per ulteriori informazioni, consulta [Fase 4: Eliminazione di un cluster](#). La procedura seguente descrive la procedura di rimozione di una o più partizioni.

1. Accedi alla AWS Management Console e apri la console MemoryDB for Redis all'[indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. Nell'elenco dei cluster, scegliere il nome del cluster da cui rimuovere una partizione.
3. Nella scheda Frammenti e nodi, scegli Aggiungi/Elimina frammenti
4. In Nuovo numero di frammenti, inserisci il numero di frammenti che desideri (con un minimo di 1).
5. Scegli Conferma per mantenere le modifiche o Annulla per annullare.

Rimozione delle partizioni (AWS CLI)

La procedura seguente descrive come riconfigurare le partizioni nel cluster MemoryDB rimuovendo le partizioni che utilizzano la AWS CLI.

⚠ Important

Prima di rimuovere le partizioni dal cluster, MemoryDB verifica che tutti i dati siano adatti alle partizioni rimanenti. Se i dati sono adatti, le partizioni vengono eliminate dal cluster come richiesto e i loro spazi chiave vengono mappati nelle partizioni rimanenti. Se i dati non rientrano negli shard rimanenti, il processo viene terminato e il cluster viene lasciato con la stessa configurazione di shard di prima della richiesta.

Puoi utilizzare la AWS CLI per rimuovere una o più partizioni dal cluster MemoryDB. Non puoi rimuovere tutti le partizioni in un cluster. Dovrai invece eliminare il cluster. Per ulteriori informazioni, consulta [Fase 4: Eliminazione di un cluster](#).

Utilizzare i seguenti parametri con `update-cluster`.

Parametri

- `--cluster-name` – Obbligatorio. Specifica su quale cluster (cluster) deve essere eseguita l'operazione di riconfigurazione dello shard.
- `--shard-configuration` – Obbligatorio. Consente di impostare il numero di frammenti utilizzando la `ShardCount` proprietà:

`ShardCount`: Imposta questa proprietà per specificare il numero di partizioni desiderato.

Example

L'esempio seguente modifica il numero di frammenti nel cluster `portandolomy-cluster` a 2.

Per Linux, macOS o Unix:

```
aws memorydb update-cluster \  
  --cluster-name my-cluster \  
  --shard-configuration \  
    ShardCount=2
```

Per Windows:

```
aws memorydb update-cluster ^
```

```
--cluster-name my-cluster ^
--shard-configuration ^
    ShardCount=2
```

Restituisce la risposta JSON seguente:

```
{
  "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
  }
}
```

Per visualizzare i dettagli del cluster aggiornato una volta che il suo stato cambia da aggiornamento a disponibile, usa il comando seguente:

Per Linux, macOS o Unix:

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

Per Windows:

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

Restituirà la seguente risposta 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
  }
]
}

```

Per ulteriori informazioni, consulta [update-cluster](#) nel Riferimento aiAWS CLI comandi.

Rimozione di partizioni (API MemoryDb)

Puoi utilizzare l'API MemoryDB per riconfigurare le partizioni nel tuo cluster MemoryDB online utilizzando l'UpdateClusteroperazione.

La procedura seguente descrive come riconfigurare le partizioni nel cluster MemoryDB rimuovendo le partizioni che utilizzano l'API MemoryDB.

Important

Prima di rimuovere le partizioni dal cluster, MemoryDB verifica che tutti i dati siano adatti alle partizioni rimanenti. Se i dati sono adatti, le partizioni vengono eliminati dal cluster come richiesto e i loro spazi chiave vengono mappati nelle partizioni rimanenti. Se i dati non rientrano negli shard rimanenti, il processo viene terminato e il cluster viene lasciato con la stessa configurazione di shard di prima della richiesta.

Puoi utilizzare l'API di MemoryDB per rimuovere una o più partizioni dal cluster MemoryDB. Non puoi rimuovere tutti le partizioni in un cluster. Dovrai invece eliminare il cluster. Per ulteriori informazioni, consulta [Fase 4: Eliminazione di un cluster](#).

Utilizzare i seguenti parametri con UpdateCluster.

Parametri

- **ClusterName** – Obbligatorio. Specifica su quale cluster (cluster) deve essere eseguita l'operazione di riconfigurazione dello shard.

- `ShardConfiguration` – Obbligatorio. Consente di impostare il numero di frammenti utilizzando la `ShardCount` proprietà:

`ShardCount`: Imposta questa proprietà per specificare il numero di partizioni desiderato.

Ridimensionamento verticale online tramite la modifica del tipo di nodo

Utilizzando il dimensionamento verticale online con MemoryDB, puoi dimensionare il cluster in modo dinamico con tempi di inattività minimi. Ciò consente al cluster di soddisfare le richieste anche durante la scalabilità.

Note

Il dimensionamento non è supportato tra un cluster di tiering di dati (ad esempio, un cluster che utilizza un tipo di nodo `r6gd`) e un cluster che non utilizza il tiering di dati (ad esempio, un cluster che utilizza un tipo di nodo `r6g`). Per ulteriori informazioni, consulta [Tiering di dati](#).

Puoi eseguire le operazioni indicate di seguito:

- Dimensionamento verso l'alto: Incrementa la capacità di lettura e scrittura modificando il tipo di nodo del cluster MemoryDB affinché utilizzi un tipo di nodo più grande.

MemoryDB ridimensiona dinamicamente il cluster rimanendo online e rispondendo alle richieste.

- Riduzione verticale - Riduce la capacità di lettura e scrittura modificando il tipo di nodo affinché utilizzi un nodo più piccolo. Di nuovo, MemoryDB ridimensiona dinamicamente il cluster rimanendo online e rispondendo alle richieste. In questo caso, il ridimensionamento del nodo permette di ridurre i costi.

Note

I processi di dimensionamento verso l'alto e il basso si basano sulla creazione di cluster con i nuovi tipi di nodo selezionati e sulla sincronizzazione dei nuovi nodi con quelli precedenti. Per garantire un'operazione di dimensionamento verso l'alto/il basso senza intoppi, procedi come segue:

- Anche se il processo di ridimensionamento verticale è progettato affinché il cluster rimanga completamente online, esso si basa sulla sincronizzazione dei dati tra il vecchio nodo e

il nuovo nodo. Si consiglia di avviare il processo di dimensionamento verso l'alto/il basso durante le ore in cui si prevede che il traffico dati sia al minimo.

- Se possibile, testa il comportamento della tua applicazione durante il ridimensionamento in un ambiente di prova.

Dimensionamento verso l'alto online

Argomenti

- [Dimensionamento verso l'alto di cluster di MemoryDB \(Console\)](#)
- [Scalabilità dei cluster MemoryDB \(AWSCLI\)](#)
- [Scalabilità dei cluster MemoryDB \(API MemoryDB\)](#)

Dimensionamento verso l'alto di cluster di MemoryDB (Console)

La procedura seguente descrive il modo in cui eseguire il dimensionamento verso l'alto di un cluster MemoryDB mediante l'AWS Management Console. Durante questo processo, il cluster MemoryDB continuerà a soddisfare le richieste con tempi di inattività minimi.

Per aumentare un cluster (console)

1. Accedi a AWS Management Console e apri la console MemoryDB for Redis all'[indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. Nell'elenco dei cluster, scegliere quello da ridimensionare.
3. Scegliere Actions (Operazioni), quindi Modify (Modifica).
4. Nella finestra di dialogo Modifica cluster:
 - Scegliere il tipo di nodo a cui dimensionare dall'elenco Node type (Tipo di nodo). Per aumentare, scegliere un tipo di nodo più grande del nodo esistente.
5. Scegliere Save changes (Salva modifiche).

Lo stato del cluster cambia in modifica. Quando lo stato cambia in disponibile, la modifica è completa ed è possibile iniziare a utilizzare il nuovo cluster.

Scalabilità dei cluster MemoryDB (AWSCLI)

La procedura seguente descrive il modo in cui eseguire il dimensionamento verso l'alto di un cluster MemoryDb mediante l'AWS CLI. Durante questo processo, il cluster MemoryDB continuerà a soddisfare le richieste con tempi di inattività minimi.

Per aumentare un cluster di MemoryDb (AWSCLI)

1. Stabilire per quali tipi di nodi è possibile eseguire il dimensionamento verso l'alto eseguendo il comando `list-allowed-node-type-updates` dell'AWS CLI con il seguente parametro.

Per Linux, macOS o Unix:

```
aws memorydb list-allowed-node-type-updates \  
  --cluster-name my-cluster-name
```

Per Windows:

```
aws memorydb list-allowed-node-type-updates ^  
  --cluster-name my-cluster-name
```

L'output del comando in alto è simile al seguente (in formato JSON).

```
{  
  "ScaleUpNodeTypes": [  
    "db.r6g.2xlarge",  
    "db.r6g.large"  
  ],  
  "ScaleDownNodeTypes": [  
    "db.r6g.large"  
  ],  
}
```

Per ulteriori informazioni, vedere [list-allowed-node-type-updates](#) nella Guida AWS CLI di riferimento.

2. Modifica il cluster per aumentare al nuovo tipo di nodo più grande, utilizzando il `AWS CLI update-cluster` comando e i seguenti parametri.
 - `--cluster-name`: Il nome del cluster che si sta ridimensionando verso l'alto.

- `--node-type`: Il nuovo tipo di nodo con il quale intendi ridimensionare il cluster. Questo valore deve essere uno dei tipi di nodi restituiti dal comando `list-allowed-node-type-updates` nella fase 1.

Per Linux, macOS o Unix:

```
aws memorydb update-cluster \  
  --cluster-name my-cluster \  
  --node-type db.r6g.2xlarge
```

Per Windows:

```
aws memorydb update-cluster ^  
  --cluster-name my-cluster ^  
  --node-type db.r6g.2xlarge ^
```

Per ulteriori informazioni, consulta [update-cluster](#).

Scalabilità dei cluster MemoryDB (API MemoryDB)

La procedura seguente ridimensiona il cluster dal tipo di nodo corrente a un nuovo tipo più grande di nodo utilizzando l'API MemoryDB. Durante questo processo, MemoryDB Aggiorna le voci DNS in modo che puntino ai nuovi nodi. Puoi dimensionare i cluster abilitati al failover automatico mentre il cluster continua a rimanere online e servire le richieste in entrata.

Il tempo necessario per aumentare a un tipo di nodo più grande varia a seconda del tipo di nodo e della quantità di dati nel cluster corrente.

Per aumentare un cluster MemoryDB (API MemoryDB)

1. Stabilire quali tipi di nodo è possibile aumentare mediante l'`ListAllowedNodeTypeUpdates` operazione API MemoryDB con il seguente parametro.
 - `ClusterName`— il nome del cluster. Utilizzate questo parametro per descrivere un cluster specifico anziché tutti i cluster.

```
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>
```

Per ulteriori informazioni, consulta la [ListAllowedNodeTypeUpdates](#) Guida di riferimento all'API MemoryDB for Redis.

2. Aumenta le dimensioni del cluster corrente fino al nuovo tipo di nodo utilizzando l'operazione `APIUpdateCluster` MemoryDB e i seguenti parametri.
 - `ClusterName`— il nome del cluster.
 - `NodeType`: Il nuovo tipo di nodo più grande di cluster in questo cluster. Questo valore deve essere uno dei tipi di istanza restituiti dall'operazione `ListAllowedNodeTypeUpdates` nella fase 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>
```

Per ulteriori informazioni, vedere [UpdateCluster](#).

Dimensionamento verso il basso online

Argomenti

- [Dimensionamento verso il basso di cluster di MemoryDB \(Console\)](#)

- [Ridimensionamento dei cluster MemoryDB \(AWSCLI\)](#)
- [Ridimensionamento dei cluster MemoryDB \(API MemoryDB\)](#)

Dimensionamento verso il basso di cluster di MemoryDB (Console)

La procedura seguente descrive il modo in cui eseguire il dimensionamento verso il basso di un cluster MemoryDb mediante l'AWS Management Console. Durante questo processo, il cluster MemoryDB continuerà a soddisfare le richieste con tempi di inattività minimi.

Per la riduzione di un cluster MemoryDB (console)

1. Accedi AWS Management Console e apri la console MemoryDB for Redis all'[indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. Nell'elenco dei cluster, scegliere quello da ridimensionare.
3. Scegliere Actions (Operazioni), quindi Modify (Modifica).
4. Nella finestra di dialogo Modifica cluster:
 - Scegliere il tipo di nodo a cui dimensionare dall'elenco Node type (Tipo di nodo). Per la riduzione verticale, scegliere un tipo di nodo più piccolo del nodo esistente. Si noti che non tutti i tipi di nodo sono disponibili per il dimensionamento.
5. Scegliere Save changes (Salva modifiche).

Lo stato del cluster cambia in modifica. Quando lo stato cambia in disponibile, la modifica è completa ed è possibile iniziare a utilizzare il nuovo cluster.

Ridimensionamento dei cluster MemoryDB (AWSCLI)

La procedura seguente descrive il modo in cui eseguire il dimensionamento verso il basso di un cluster MemoryDb mediante l'AWS CLI. Durante questo processo, il cluster MemoryDB continuerà a soddisfare le richieste con tempi di inattività minimi.

Per la riduzione di un cluster di MemoryDb (AWSCLI)

1. Stabilire per quali tipi di nodi è possibile eseguire il ridimensionamento verso il basso eseguendo il comando AWS CLI `list-allowed-node-type-updates` con il seguente parametro.

Per Linux, macOS o Unix:

```
aws memorydb list-allowed-node-type-updates \  
  --cluster-name my-cluster-name
```

Per Windows:

```
aws memorydb list-allowed-node-type-updates ^\  
  --cluster-name my-cluster-name
```

L'output del comando in alto è simile al seguente (in formato JSON).

```
{  
  "ScaleUpNodeTypes": [  
    "db.r6g.2xlarge",  
    "db.r6g.large"  
  ],  
  "ScaleDownNodeTypes": [  
    "db.r6g.large"  
  ],  
}
```

Per ulteriori informazioni, vedere [list-allowed-node-type-updates](#).

2. Modifica il tuo cluster per ridurlo al nuovo tipo di nodo più piccolo, utilizzando il `update-cluster` comando e i seguenti parametri.
 - `--cluster-name`: Il nome del cluster che si sta ridimensionando verso il basso.
 - `--node-type`: Il nuovo tipo di nodo con il quale intendi ridimensionare il cluster. Questo valore deve essere uno dei tipi di nodi restituiti dal comando `list-allowed-node-type-updates` nella fase 1.

Per Linux, macOS o Unix:

```
aws memorydb update-cluster \  
  --cluster-name my-cluster \  
  --node-type db.r6g.large
```

Per Windows:

```
aws memorydb update-cluster ^  
  --cluster-name my-cluster ^  
  --node-type db.r6g.large
```

Per ulteriori informazioni, consulta [update-cluster](#).

Ridimensionamento dei cluster MemoryDB (API MemoryDB)

La procedura seguente ridimensiona il cluster dal tipo di nodo corrente a un nuovo tipo di nodo di minori dimensioni utilizzando l'API MemoryDB. Durante questo processo, il cluster MemoryDB continuerà a soddisfare le richieste con tempi di inattività minimi.

Il tempo necessario per diminuire utilizzando un tipo di nodo di minori dimensioni varia a seconda del tipo di nodo e della quantità di dati nel cluster corrente.

Riduzione verticale (API di MemoryDb)

1. Stabilire quali tipi di nodo è possibile diminuire mediante l'[ListAllowedNodeTypeUpdates](#) API con il seguente parametro:
 - `ClusterName`— il nome del cluster. Utilizzate questo parametro per descrivere un cluster specifico anziché tutti i cluster.

```
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. Dimensionare il cluster corrente affinché utilizzi il nuovo tipo di nodo utilizzando l'[UpdateCluster](#) API con i seguenti parametri.
 - `ClusterName`— il nome del cluster.

- **NodeType**: Il nuovo tipo di nodo più piccolo da usare nei cluster di questo cluster. Questo valore deve essere uno dei tipi di istanza restituiti dall'operazione `ListAllowedNodeTypeUpdates` nella fase 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>
```

Configurazione dei parametri di motore con i gruppi di parametri

MemoryDB for Redis utilizza i parametri per controllare le proprietà di runtime dei nodi e dei cluster. Le versioni più recenti del prodotto includono in genere parametri aggiuntivi per il support delle nuove funzionalità. Per le tabelle dei parametri, consulta [Parametri specifici di Redis](#).

Come previsto, alcuni valori di parametro, ad esempio `maxmemory`, sono determinati da tipo di nodo e motore. Per una tabella di questi valori di parametro per tipo di nodo, consulta [Parametri specifici del tipo di nodo di MemoryDB](#).

Argomenti

- [Gestione dei parametri](#)
- [Livelli dei gruppi di parametri](#)
- [Creazione di un gruppo di parametri](#)
- [Elenco di gruppi di parametri per nome](#)
- [Generazione di un elenco di valori di un gruppo di parametri](#)
- [Modifica di un gruppo di parametri](#)

- [Eliminazione di un gruppo di parametri](#)
- [Parametri specifici di Redis](#)

Gestione dei parametri

Ai fini di semplificarne la gestione, i parametri sono raggruppati in gruppi di parametri denominati. Un gruppo di parametri rappresenta una combinazione di valori specifici per i parametri passati al software del motore durante l'avvio. Questi valori determinano il comportamento dei processi del motore su ciascun nodo in fase di runtime. I valori dei parametri su un gruppo di parametri specifico si applicano a tutti i nodi associati al gruppo, indipendentemente dal cluster a cui appartengono.

Per ottimizzare le prestazioni del cluster, puoi modificare alcuni valori dei parametri oppure puoi modificare il gruppo di parametri del cluster.

- Non è possibile modificare né eliminare i gruppi di parametri predefiniti. Se hai bisogno di valori dei parametri personalizzati, devi creare un gruppo di parametri personalizzato.
- La famiglia del gruppo di parametri e il cluster che assegna devono essere compatibili. Ad esempio, se sul cluster è in esecuzione la versione 6 di Redis, è possibile utilizzare solo gruppi di parametri, predefiniti o personalizzati, della famiglia `memorydb_redis6`.
- Quando modifichi i parametri di un cluster, la modifica viene applicata al cluster immediatamente. Ciò è valido se modifichi il gruppo dei parametri del cluster o un valore di parametro nel gruppo dei parametri del cluster.

Livelli dei gruppi di parametri

Livelli di gruppi di parametri MemoryDB for Redis

Di default globale

Il gruppo di parametri root di primo livello per tutti i clienti di MemoryDB for Redis della regione.

Il gruppo globale di parametri predefinito:

- È riservato a MemoryDB e non è disponibile per il cliente.

Di default del cliente

Una copia del gruppo di parametri Global Default creato per l'utilizzo da parte del cliente.

Il gruppo di parametri Customer Default:

- È creato e di proprietà di MemoryDB.
- È disponibile al cliente per l'uso come gruppo di parametri per tutti i cluster che utilizzano una versione del motore supportata da questo gruppo di parametri.
- Non può essere modificato dal cliente.

Di proprietà del cliente

Una copia del gruppo di parametri Customer Default. Un gruppo di parametri Customer Owned viene creato ogni volta che il cliente crea un gruppo di parametri.

Il gruppo di parametri Customer Owned:

- Viene creato dal cliente ed è di sua proprietà.
- Può essere assegnato a un cluster compatibile del cliente.
- Può essere modificato dal cliente per creare un gruppo di parametri personalizzato.

Non è possibile modificare tutti i valori dei parametri. Per ulteriori informazioni, consulta [Parametri specifici di Redis](#).

Creazione di un gruppo di parametri

Devi creare un nuovo gruppo di parametri se per uno o più valori di parametri desideri configurare un'impostazione diversa da quella predefinita. È possibile creare un gruppo di parametri utilizzando la console MemoryDB AWS CLI, o l'API MemoryDB.

Creazione di un gruppo di parametri (Console)

La procedura seguente mostra come creare un gruppo di parametri utilizzando la console MemoryDB.

Per creare un gruppo di parametri utilizzando la console MemoryDB

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Per un elenco di tutti i gruppi di parametri disponibili, nel pannello di navigazione a sinistra scegliere Parameter Groups (Gruppi di parametri).
3. Per creare un gruppo di parametri, scegli Crea gruppo di parametri.

Viene visualizzata la pagina Crea gruppo di parametri.

4. Nella casella Name (Nome) digitare un nome univoco per il gruppo di parametri.

Quando si crea un cluster o si modifica un gruppo di parametri del cluster, il gruppo di parametri viene scelto in base al relativo nome. È pertanto consigliabile che il nome sia informativo e identifichi in qualche modo la famiglia del gruppo di parametri.

I vincoli per la denominazione dei gruppi di parametri sono i seguenti:

- Devono iniziare con una lettera ASCII.
 - Può contenere solo lettere ASCII, cifre e trattini ('-').
 - Deve contenere da 1 a 255 caratteri.
 - Non possono contenere due trattini consecutivi.
 - Non posso terminare con un trattino.
5. Nella casella Description (Descrizione) digitare una descrizione per il gruppo di parametri.
 6. Nella casella Compatibilità della versione Redis, scegli una versione del motore a cui corrisponde questo gruppo di parametri.

7. Nella sezione Tag, è possibile aggiungere tag per cercare e filtrare i gruppi di parametri o tenere traccia AWS dei costi.
8. Per creare il gruppo di parametri, scegliere Create (Crea).

Per terminare il processo senza creare il gruppo di parametri, scegliere Cancel (Annulla).

9. Quando viene creato, il gruppo di parametri è associato ai valori predefiniti della famiglia. Per modificare i valori predefiniti, è necessario modificare il gruppo di parametri. Per ulteriori informazioni, consulta [Modifica di un gruppo di parametri](#).

Creazione di un gruppo di parametri (AWS CLI)

Per creare un gruppo di parametri utilizzando il AWS CLI, utilizzate il comando `create-parameter-group` con questi parametri.

- `--parameter-group-name` - Nome del gruppo di parametri.

Vincoli per la denominazione dei gruppi di parametri:

- Devono iniziare con una lettera ASCII.
- Può contenere solo lettere ASCII, cifre e trattini ('-').
- Deve contenere da 1 a 255 caratteri.
- Non possono contenere due trattini consecutivi.
- Non posso terminare con un trattino.
- `--family`— Il motore e la famiglia di versioni per il gruppo di parametri.
- `--description` - Una descrizione per la copia del gruppo di parametri del cluster fornita dall'utente.

Example

L'esempio seguente crea un gruppo di parametri denominato `MyRedis6x` utilizzando la famiglia `memorydb_redis6` come modello.

Per Linux, macOS o Unix:

```
aws memorydb create-parameter-group \  
  --parameter-group-name myRedis6x \  
  --family memorydb_redis6 \  
  --tags myRedis6x \  
  --description MyRedis6x
```

```
--description "My first parameter group"
```

Per Windows:

```
aws memorydb create-parameter-group ^
  --parameter-group-name myRedis6x ^
  --family memorydb_redis6 ^
  --description "My first parameter group"
```

L'output di questo comando dovrebbe essere simile a quanto segue:

```
{
  "ParameterGroup": {
    "Name": "myRedis6x",
    "Family": "memorydb_redis6",
    "Description": "My first parameter group",
    "ARN": "arn:aws:memorydb:us-east-1:012345678912:parametergroup/myredis6x"
  }
}
```

Quando viene creato, il gruppo di parametri è associato ai valori predefiniti della famiglia. Per modificare i valori predefiniti, è necessario modificare il gruppo di parametri. Per ulteriori informazioni, consulta [Modifica di un gruppo di parametri](#).

Per ulteriori informazioni, consultare [create-parameter-group](#).

Creazione di un gruppo di parametri (API MemoryDB)

Per creare un gruppo di parametri utilizzando l'API MemoryDB, utilizzate l'CreateParameterGroupazione con questi parametri.

- ParameterGroupName - Nome del gruppo di parametri.

Vincoli per la denominazione dei gruppi di parametri:

- Devono iniziare con una lettera ASCII.
- Può contenere solo lettere ASCII, cifre e trattini ('-').
- Deve contenere da 1 a 255 caratteri.
- Non possono contenere due trattini consecutivi.
- Non posso terminare con un trattino.

- **Family**— Il motore e la famiglia di versioni per il gruppo di parametri. Ad esempio, `memorydb_redis6`.
- **Description** - Una descrizione per la copia del gruppo di parametri del cluster fornita dall'utente.

Example

L'esempio seguente crea un gruppo di parametri denominato `MyRedis6x` utilizzando la famiglia `memorydb_redis6` come modello.

```
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 risposta restituita da tale operazione dovrebbe essere simile a quanto segue:

```
<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>
```

Quando viene creato, il gruppo di parametri è associato ai valori predefiniti della famiglia. Per modificare i valori predefiniti, è necessario modificare il gruppo di parametri. Per ulteriori informazioni, consulta [Modifica di un gruppo di parametri](#).

Per ulteriori informazioni, consultare [CreateParameterGroup](#).

Elenco di gruppi di parametri per nome

È possibile elencare i gruppi di parametri utilizzando la console MemoryDB, o l'API MemoryDB. AWS CLI

Elenco di gruppi di parametri per nome (console)

La procedura seguente mostra come visualizzare un elenco dei gruppi di parametri utilizzando la console MemoryDB.

Per elencare i gruppi di parametri utilizzando la console MemoryDB

1. [Accedere AWS Management Console e aprire la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Per un elenco di tutti i gruppi di parametri disponibili, nel pannello di navigazione a sinistra scegliere Parameter Groups (Gruppi di parametri).

Elenco dei gruppi di parametri per nome (AWS CLI)

Per generare un elenco di gruppi di parametri utilizzando il AWS CLI, utilizzare il comando `describe-parameter-groups`. Se specifichi un nome del gruppo di parametri, nell'elenco sarà presente solo tale gruppo di parametri. Se non specifichi un nome del gruppo di parametri, nell'elenco saranno presenti fino a `--max-results` gruppi di parametri. In entrambi i casi, saranno indicati nome, famiglia e descrizione del gruppo di parametri.

Example

Il codice di esempio seguente elenca il gruppo di parametri `MyRedis6x`.

Per Linux, macOS o Unix:

```
aws memorydb describe-parameter-groups \  
  --parameter-group-name myRedis6x
```

Per Windows:

```
aws memorydb describe-parameter-groups ^  
  --parameter-group-name myRedis6x
```


L'output di questo comando sarà simile al seguente e conterrà il nome, la famiglia e la descrizione del gruppo di parametri.

```
{
  "ParameterGroups": [
    {
      "Name": "myRedis6x",
      "Family": "memorydb_redis6",
      "Description": "My first parameter group",
      "ARN": "arn:aws:memorydb:us-east-1:012345678912:parametergroup/
myredis6x"
    }
  ]
}
```

Example

Il codice di esempio seguente elenca il gruppo di parametri myRedis6x per i gruppi di parametri in esecuzione sul motore Redis dalla versione 5.0.6 in poi.

Per Linux, macOS o Unix:

```
aws memorydb describe-parameter-groups \
  --parameter-group-name myRedis6x
```

Per Windows:

```
aws memorydb describe-parameter-groups ^
  --parameter-group-name myRedis6x
```

L'output di questo comando sarà simile al seguente, con l'elenco del nome, della famiglia e della descrizione del gruppo di parametri.

```
{
  "ParameterGroups": [
    {
      "Name": "myRedis6x",
      "Family": "memorydb_redis6",
      "Description": "My first parameter group",
      "ARN": "arn:aws:memorydb:us-east-1:012345678912:parametergroup/
myredis6x"
    }
  ]
}
```

```
    }  
  ]  
}
```

Example

Il codice di esempio seguente elenca fino a 20 gruppi di parametri.

```
aws memorydb describe-parameter-groups --max-results 20
```

L'output JSON di questo comando sarà simile al seguente, con l'elenco del nome, della famiglia e della descrizione di ogni gruppo di parametri.

```
{  
  "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"  
    },  
    ...  
  ]  
}
```

Per ulteriori informazioni, consulta [describe-parameter-groups](#).

Elenco dei gruppi di parametri per nome (API MemoryDB)

Per generare un elenco di gruppi di parametri utilizzando l'API MemoryDB, utilizzate l'azione `DescribeParameterGroups`. Se specificate un nome del gruppo di parametri, nell'elenco sarà presente solo tale gruppo di parametri. Se non specificate un nome del gruppo di parametri, nell'elenco saranno presenti fino a `MaxResults` gruppi di parametri. In entrambi i casi, saranno indicati nome, famiglia e descrizione del gruppo di parametri.

Example

Il codice di esempio seguente elenca fino a 20 gruppi di parametri.

```
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 risposta di questa azione sarà simile a questa, con l'elenco del nome, della famiglia e della descrizione nel caso di `memorydb_redis6`, per ogni gruppo di parametri.

```
<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

Il codice di esempio seguente elenca il gruppo di parametri `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 risposta restituita da tale operazione sarà simile a quanto segue e conterrà il nome, la famiglia e la descrizione.

```
<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>
```

Per ulteriori informazioni, consulta [DescribeParameterGroups](#).

Generazione di un elenco di valori di un gruppo di parametri

È possibile elencare i parametri e i relativi valori per un gruppo di parametri utilizzando la console MemoryDB, l'API MemoryDB o l' AWS CLI API MemoryDB.

Generazione di un elenco di valori di un gruppo di parametri (console)

La procedura seguente mostra come elencare i parametri e i relativi valori per un gruppo di parametri utilizzando la console MemoryDB.

Per elencare i parametri di un gruppo di parametri e i relativi valori utilizzando la console MemoryDB

1. [Accedere AWS Management Console e aprire la console MemoryDB for Redis all'indirizzo `https://console.aws.amazon.com/memorydb/`.](https://console.aws.amazon.com/memorydb/)
2. Per un elenco di tutti i gruppi di parametri disponibili, nel pannello di navigazione a sinistra scegliere Parameter Groups (Gruppi di parametri).
3. Scegliete il gruppo di parametri per il quale desiderate elencare i parametri e i valori scegliendo il nome (non la casella accanto) del nome del gruppo di parametri.

I parametri e i relativi valori verranno elencati nella parte inferiore dello schermo. A causa dell'elevato numero di parametri, potrebbe essere necessario scorrere verso l'alto e verso il basso per individuare il parametro desiderato.

Elenco dei valori di un gruppo di parametri (AWS CLI)

Per elencare i parametri di un gruppo di parametri e i relativi valori utilizzando il AWS CLI, utilizzare il comandodescribe-parameters.

Example

Il codice di esempio seguente elenca tutti i parametri e i relativi valori per il gruppo di parametri MyRedis6x.

Per Linux, macOS o Unix:

```
aws memorydb describe-parameters \  
  --parameter-group-name myRedis6x
```

Per Windows:

```
aws memorydb describe-parameters ^  
  --parameter-group-name myRedis6x
```

Per ulteriori informazioni, consulta [describe-parameters](#).

Elenco dei valori di un gruppo di parametri (API MemoryDB)

Per elencare i parametri di un gruppo di parametri e i relativi valori utilizzando l'API MemoryDB, utilizzate l'azione `DescribeParameters`.

Per ulteriori informazioni, consulta [DescribeParameters](#).

Modifica di un gruppo di parametri

Important

Non è consentito modificare un gruppo di parametri di default.

Non puoi modificare alcuni valori di parametri in un gruppo di parametri. Tali valori di parametri sono applicati ai cluster associati al gruppo di parametri. Per ulteriori informazioni su quando una modifica dei valori di parametri viene applicata a un gruppo di parametri, consulta [Parametri specifici di Redis](#).

Modifica di un gruppo di parametri (console)

La procedura seguente mostra come modificare il valore del parametro utilizzando la console MemoryDB. Puoi usare la stessa procedura per modificare il valore di qualsiasi parametro.

Per modificare il valore di un parametro utilizzando la console MemoryDB

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. Per un elenco di tutti i gruppi di parametri disponibili, nel pannello di navigazione a sinistra scegliere **Parameter Groups** (Gruppi di parametri).
3. Scegli il gruppo di parametri che desideri modificare facendo clic sul pulsante di opzione a sinistra del nome del gruppo di parametri.

Scegli **Azioni**, quindi **Visualizza dettagli**. In alternativa, puoi anche scegliere il nome del gruppo di parametri per accedere alla pagina dei dettagli.

4. Per modificare il parametro, scegliete Modifica. Tutti i parametri modificabili potranno essere modificati. Potrebbe essere necessario spostarsi tra le pagine per trovare il parametro che si desidera modificare. In alternativa, puoi cercare il parametro per nome, valore o tipo nella casella di ricerca.
5. Apportate le modifiche necessarie ai parametri.
6. Per salvare le modifiche, scegliere Save changes (Salva modifiche).
7. Se hai modificato i valori dei parametri su un numero di pagine, puoi rivedere tutte le modifiche scegliendo Antepima modifiche. Per confermare le modifiche, scegliete Salva modifiche. Per apportare altre modifiche, scegli Indietro.
8. La pagina dei dettagli dei parametri offre anche la possibilità di ripristinare i valori predefiniti. Per ripristinare i valori predefiniti, scegliete Ripristina valori predefiniti. Le caselle di controllo appariranno sul lato sinistro di tutti i parametri. Puoi selezionare quelli che desideri ripristinare e scegliere Procedi al ripristino per confermare.

Scegli conferma per confermare l'azione di ripristino nella finestra di dialogo.

9. La pagina dei dettagli dei parametri consente di impostare il numero di parametri che si desidera visualizzare su ciascuna pagina. Usa la ruota dentata sul lato destro per apportare queste modifiche. Puoi anche abilitare/disabilitare le colonne che desideri nella pagina dei dettagli. Queste modifiche durano per tutta la sessione della console.

Per trovare il nome del parametro modificato, consultare [Parametri specifici di Redis](#).

Modifica di un gruppo di parametri (AWS CLI)

Per modificare il valore di un parametro utilizzando il AWS CLI, utilizzate il comando. `update-parameter-group`

Per individuare il nome e i valori consentiti del parametro da modificare, consulta [Parametri specifici di Redis](#)

Per ulteriori informazioni, vedere [update-parameter-group](#).

Modifica di un gruppo di parametri (API MemoryDB)

Per modificare i valori dei parametri di un gruppo di parametri utilizzando l'API MemoryDB, utilizzate l'azione. `UpdateParameterGroup`

Per individuare il nome e i valori consentiti del parametro da modificare, consulta [Parametri specifici di Redis](#)

Per ulteriori informazioni, consulta [UpdateParameterGroup](#).

Eliminazione di un gruppo di parametri

È possibile eliminare un gruppo di parametri personalizzato utilizzando la console MemoryDB AWS CLI, o l'API MemoryDB.

Un gruppo di parametri non può essere eliminato se è associato a cluster. Non è inoltre possibile eliminare i gruppi di parametri predefiniti.

Eliminazione di un gruppo di parametri (console)

La procedura seguente mostra come eliminare un gruppo di parametri utilizzando la console MemoryDB.

Per eliminare un gruppo di parametri utilizzando la console MemoryDB

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Per un elenco di tutti i gruppi di parametri disponibili, nel pannello di navigazione a sinistra scegliere Parameter Groups (Gruppi di parametri).
3. Scegli i gruppi di parametri che desideri eliminare facendo clic sul pulsante di opzione a sinistra del nome del gruppo di parametri.

Scegliere Actions (Operazioni), quindi selezionare Delete (Elimina VPC).

4. Verrà visualizzata la schermata di conferma Delete Parameter Groups (Elimina gruppi di parametri).
5. Per eliminare i gruppi di parametri, inserite Elimina nella casella di testo di conferma.

Per mantenere i gruppi di parametri, scegliere Cancel (Annulla).

Eliminazione di un gruppo di parametri (AWS CLI)

Per eliminare un gruppo di parametri utilizzando il AWS CLI, utilizzate il comando `delete-parameter-group` Per il gruppo di parametri da eliminare, il gruppo di parametri specificato da `--parameter-group-name` non può essere associato ad alcun cluster, né può essere un gruppo di parametri di default.

Il codice di esempio seguente elimina il gruppo di parametri MyRedis6x.

Example

Per Linux, macOS o Unix:

```
aws memorydb delete-parameter-group \  
  --parameter-group-name myRedis6x
```

Per Windows:

```
aws memorydb delete-parameter-group ^  
  --parameter-group-name myRedis6x
```

Per ulteriori informazioni, vedere [delete-parameter-group](#)

Eliminazione di un gruppo di parametri (API MemoryDB)

Per eliminare un gruppo di parametri utilizzando l'API MemoryDB, utilizzate l'azione.

DeleteParameterGroup Per il gruppo di parametri da eliminare, il gruppo di parametri specificato da ParameterGroupName non può essere associato ad alcun cluster, né può essere un gruppo di parametri di default.

Example

Il codice di esempio seguente elimina il gruppo di parametri 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>
```

Per ulteriori informazioni, consulta [DeleteParameterGroup](#).

Parametri specifici di Redis

Se non specifichi un gruppo di parametri per il cluster Redis, verrà utilizzato un gruppo di parametri di default appropriato per la versione del motore. In un gruppo di parametri di default non puoi modificare i valori di nessuno dei parametri. Tuttavia puoi creare un gruppo di parametri personalizzato e assegnarlo in qualsiasi momento al cluster, purché i valori dei parametri modificabili in base a condizioni corrispondano in entrambi i gruppi di parametri. Per ulteriori informazioni, consulta [Creazione di un gruppo di parametri](#).

Argomenti

- [Modifiche dei parametri di Redis 7](#)
- [Parametri Redis 6](#)
- [Parametri specifici del tipo di nodo di MemoryDB](#)

Modifiche dei parametri di Redis 7

Note

MemoryDB ha introdotto una versione di anteprima di [Vector search](#) che include un nuovo gruppo di parametri immutabili. `default.memorydb-redis7.search.preview` Questo gruppo di parametri è disponibile nella console di MemoryDB e durante la creazione di un nuovo `vector-search-enabled` cluster utilizzando il comando CLI [create-cluster](#). La versione di anteprima è disponibile nelle seguenti AWS regioni: Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (Oregon), Asia Pacifico (Tokyo) ed Europa (Irlanda).

Famiglia di gruppi di parametri: `memorydb_redis7`

I parametri aggiunti in Redis 7 sono i seguenti.

Nome	Informazioni	Descrizione
<code>latency-tracking</code>	Valori consentiti: <code>yes</code> , <code>no</code> Impostazione predefinita: <code>no</code>	Se impostato su <code>yes</code> , tiene traccia delle latenze per comando e consente di esportare la distribuzione percentile tramite il comando delle statistiche di latenza <code>INFO</code> e le distribuzioni

Nome	Informazioni	Descrizione
	<p>Tipo: string</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente in tutti i nodi del cluster.</p>	<p>di latenza cumulative (istogrammi) tramite il comando LATENCY.</p>
<code>hash-max-listpack-entries</code>	<p>Valori consentiti: 0+</p> <p>Impostazione predefinita: 512</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente in tutti i nodi del cluster.</p>	<p>Il numero massimo di voci hash per consentire la compressione del set di dati.</p>
<code>hash-max-listpack-value</code>	<p>Valori consentiti: 0+</p> <p>Impostazione predefinita: 64</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente in tutti i nodi del cluster.</p>	<p>La soglia del numero massimo di voci hash per consentire la compressione del set di dati.</p>

Nome	Informazioni	Descrizione
<code>zset-max-listpack-entries</code>	<p>Valori consentiti: 0+</p> <p>Impostazione predefinita: 128</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente in tutti i nodi del cluster.</p>	Il numero massimo di voci set ordinari per consentire la compressione del set di dati.
<code>zset-max-listpack-value</code>	<p>Valori consentiti: 0+</p> <p>Impostazione predefinita: 64</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente in tutti i nodi del cluster.</p>	La soglia del numero massimo di voci set ordinari per consentire la compressione del set di dati.

I parametri modificati in Redis 7 sono i seguenti.

Nome	Informazioni	Descrizione
<code>activeresharding</code>	<p>Modificabile: no. In Redis 7, questo parametro è nascosto e abilitato per impostazione predefinita. Per disattivarlo, è necessario creare un caso di supporto.</p>	Era modificabile.

I parametri rimossi in Redis 7 sono i seguenti.

Nome	Informazioni	Descrizione
<code>hash-max-ziplist-entries</code>	<p>Valori consentiti: 0+</p> <p>Impostazione predefinita: 512</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente in tutti i nodi del cluster.</p>	Utilizzare <code>listpack</code> anziché <code>ziplist</code> per rappresentare la codifica hash piccola
<code>hash-max-ziplist-value</code>	<p>Valori consentiti: 0+</p> <p>Impostazione predefinita: 64</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente in tutti i nodi del cluster.</p>	Utilizzare <code>listpack</code> anziché <code>ziplist</code> per rappresentare la codifica hash piccola
<code>zset-max-ziplist-entries</code>	<p>Valori consentiti: 0+</p> <p>Impostazione predefinita: 128</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente in tutti i nodi del cluster.</p>	Utilizzare <code>listpack</code> anziché <code>ziplist</code> per rappresentare la codifica hash piccola.

Nome	Informazioni	Descrizione
<code>zset-max-ziplist-value</code>	<p>Valori consentiti: 0+</p> <p>Impostazione predefinita: 64</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente in tutti i nodi del cluster.</p>	Utilizzare <code>listpack</code> anziché <code>ziplist</code> per rappresentare la codifica hash piccola.

Parametri Redis 6

Note

Nella versione 6.2 del motore Redis, quando la famiglia di nodi `r6gd` è stata introdotta per l'uso con [Tiering di dati](#), solo le politiche di `allkeys-lru` memoria massima sono supportate con i noeviction tipi `volatile-lru` di nodi `r6gd`.

Famiglia di gruppi di parametri: `memorydb_redis6`

I parametri aggiunti in Redis 6 sono i seguenti.

Nome	Informazioni	Descrizione
<code>maxmemory-policy</code>	<p>Tipo: STRING</p> <p>Valori consentiti: <code>volatile-lru</code>, <code>allkeys-lru</code>, <code>volatile-lfu</code>, <code>allkeys-lfu</code>, <code>volatile-random</code>, <code>allkeys-random</code>, <code>volatile-ttl</code>, <code>noeviction</code></p> <p>Predefinito: <code>noeviction</code></p>	<p>La policy di espulsione per le chiavi quando viene raggiunto l'utilizzo di memoria massimo.</p> <p>Per ulteriori informazioni, vedere Utilizzo di Redis come cache LRU Utilizzo di Redis come cache LRU.</p>

Nome	Informazioni	Descrizione
list-compress-depth	Tipo: INTEGER Valori consentiti: 0- Impostazione Predefinita: 0	<p>La profondità di compressione (compress depth) è il numero di nodi quicklist ziplist di ciascun lato dell'elenco da escludere dalla compressione. I nodi head e tail dell'elenco vengono sempre decompressi per ottenere operazioni di push e pop rapide. Le impostazioni sono:</p> <ul style="list-style-type: none"> • 0: disabilita completamente la compressione. • 1: la compressione inizia con il primo nodo successivo a head e termina con il primo nodo precedente a tail. [head]->nodo->nodo->...->nodo->[tail] Vengono compressi tutti i nodi tranne [head] e [tail]. • 2: la compressione inizia con il secondo nodo successivo a head e termina con il secondo nodo precedente a tail. [head]->[succ]->nodo->nodo->...->nodo->[prec]->[tail] [head], [succ], [prec], [tail] non vengono compressi. Vengono compressi tutti gli altri nodi. • ecc.

Nome	Informazioni	Descrizione
<code>hll-spars e-max-byt es</code>	<p>Tipo: INTEGER</p> <p>Valori consentiti: 1-16000</p> <p>Impostazione predefinita: 3000</p>	<p>HyperLogLog limite di byte di rappresentazione sparsa. Il limite include l'intestazione a 16 byte. Quando si HyperLogLog utilizza la rappresentazione sparsa supera questo limite, viene convertita nella rappresentazione densa.</p> <p>Un valore maggiore di 16000 non è consigliato, perché a quel punto la rappresentazione densa è più efficiente in termini di memoria.</p> <p>Consigliamo un valore di circa 3000 per sfruttare i vantaggi della codifica efficiente in termini di spazio senza rallentamenti PFADD eccessivi, ossia $O(N)$ con la codifica sparsa. Il valore può essere aumentato a ~10000 quando la CPU non è un problema, ma lo spazio sì, e il set di dati è composto da molti dati HyperLogLog con cardinalità compresa tra 0 e 15000.</p>
<code>lfu-log-f actor</code>	<p>Tipo: INTEGER</p> <p>Valori consentiti: 1-</p> <p>Impostazione predefinita: 10</p>	<p>Il fattore logaritmico per incrementare il contatore chiave della politica di sfratto della LFU.</p>
<code>lfu-decay -time</code>	<p>Tipo: INTEGER</p> <p>Valori consentiti: 0-</p> <p>Impostazione predefinita: 1</p>	<p>La quantità di tempo, in minuti, necessaria per ridurre il contatore chiave della politica di sfratto della LFU.</p>

Nome	Informazioni	Descrizione
<code>active-defrag-max-scan-fields</code>	Tipo: INTEGER Valori consentiti: 1-1000000 Impostazione predefinita: 1000	Numero massimo di campi set/hash/zset/list che verranno elaborati dalla scansione del dizionario principale durante la deframmentazione attiva.
<code>active-defrag-threshold-upper</code>	Tipo: INTEGER Valori consentiti: 1-100 Impostazione predefinita: 100	Percentuale massima di frammentazione che richiede lo sforzo massimo.
<code>client-output-buffer-limit-pubsub-hard-limit</code>	Tipo: INTEGER Valori consentiti: 0- Impostazione predefinita: 33554432	Per i client di pubblicazione/sottoscrizione e Redis: se il buffer di output di un client raggiunge il numero di byte specificato, il client verrà disconnesso.
<code>client-output-buffer-limit-pubsub-soft-limit</code>	Tipo: INTEGER Valori consentiti: 0- Impostazione predefinita: 8388608	Per i client Redis publish/subscribe: se il buffer di output di un client raggiunge il numero di byte specificato, il client verrà disconnesso, ma solo se questa condizione persiste per <code>client-output-buffer-limit-pubsub-soft-seconds</code> .
<code>client-output-buffer-limit-pubsub-soft-seconds</code>	Tipo: INTEGER Valori consentiti: 0- Impostazione predefinita: 60	Per i client di pubblicazione/sottoscrizione Redis: se il buffer di output di un client rimane di <code>client-output-buffer-limit-pubsub-soft-limit</code> per un tempo maggiore del numero di secondi specificato, il client verrà disconnesso.

Nome	Informazioni	Descrizione
<code>timeout</code>	<p>Tipo: INTEGER</p> <p>Valori consentiti: 0,20-</p> <p>Impostazione Predefinita: 0</p>	<p>Il numero di secondi prima del timeout di un nodo. I valori sono:</p> <ul style="list-style-type: none"> • 0 — non disconnettere mai un client inattivo. • 1-19: valori non validi. • ≥ 20: il numero di secondi di attesa di un nodo prima di disconnettere un client inattivo.
<code>notify-keyspace-events</code>	<p>Tipo: STRING</p> <p>Valori consentiti: NULL</p> <p>Impostazione predefinita: NULL</p>	<p>Gli eventi keyspace su cui Redis deve informare i client Pub/Sub. Per impostazione predefinita, tutte le notifiche sono disabilitate.</p>
<code>maxmemory-samples</code>	<p>Tipo: INTEGER</p> <p>Valori consentiti: 1-</p> <p>Impostazione predefinita: 3</p>	<p>Per least-recently-used (LRU) i time-to-live (TTL) calcoli, questo parametro rappresenta la dimensione del campione di chiavi da controllare. Per impostazione predefinita, Redis sceglie 3 chiavi e utilizza quella usata meno di recente.</p>
<code>slowlog-max-len</code>	<p>Tipo: INTEGER</p> <p>Valori consentiti: 0-</p> <p>Impostazione predefinita: 128</p>	<p>La lunghezza massima di Slow Log (Log lento) di Redis. Non ci sono limiti a questa lunghezza. Tieni presente che consumerà memoria. È possibile recuperare la memoria utilizzata dallo slow log con <code>SLOWLOG RESET</code>.</p>

Nome	Informazioni	Descrizione
<code>activeresharding</code>	<p>Tipo: STRING</p> <p>Valori consentiti: sì, no</p> <p>Impostazione predefinita: yes (sì)</p>	<p>La tabella hash principale viene sottoposta a rehashing 10 volte al secondo. Ogni operazione di rehashing utilizza 1 millisecondo di tempo CPU.</p> <p>Questo valore viene impostato quando crei il gruppo di parametri. Al momento dell'assegnazione di un nuovo gruppo di parametri a un cluster, questo valore deve corrispondere in entrambi i gruppi di parametri: il precedente e il nuovo.</p>
<code>client-output-buffer-limit-normal-hard-limit</code>	<p>Tipo: INTEGER</p> <p>Valori consentiti: 0-</p> <p>Impostazione Predefinita: 0</p>	<p>Se il buffer di output di un client raggiunge il numero di byte specificato, il client verrà disconnesso. Il valore di default è zero (nessun limite rigido).</p>
<code>client-output-buffer-limit-normal-soft-limit</code>	<p>Tipo: INTEGER</p> <p>Valori consentiti: 0-</p> <p>Impostazione Predefinita: 0</p>	<p>Se il buffer di output di un client raggiunge il numero di byte specificato, il client verrà disconnesso, ma solo se questa condizione persiste per <code>client-output-buffer-limit-normal-soft-seconds</code>. Il valore di default è zero (nessun limite flessibile).</p>
<code>client-output-buffer-limit-normal-soft-seconds</code>	<p>Tipo: INTEGER</p> <p>Valori consentiti: 0-</p> <p>Impostazione Predefinita: 0</p>	<p>Se il buffer di output di un client rimane di <code>client-output-buffer-limit-normal-soft-limit</code> byte per un tempo maggiore del numero di secondi specificato, il client verrà disconnesso. Il valore di default è zero (nessun limite di tempo).</p>

Nome	Informazioni	Descrizione
<code>tcp-keepalive</code>	Tipo: INTEGER Valori consentiti: 0- Impostazione predefinita: 300	Se impostato su un valore diverso da zero (N), i client del nodo vengono sottoposti a polling ogni N secondi, per garantire che siano ancora connessi. Con l'impostazione predefinita 0, non viene eseguito alcun polling.
<code>active-defrag-cycle-min</code>	Tipo: INTEGER Valori consentiti: 1-75 Impostazione predefinita: 5	Sforzo minimo per la deframmentazione in percentuale di CPU.
<code>stream-node-max-bytes</code>	Tipo: INTEGER Valori consentiti: 0- Impostazione predefinita: 4096	La struttura dati del flusso è una struttura radice che codifica più voci al suo interno. Utilizza questa configurazione per specificare le dimensioni massime in byte di un singolo nodo in una struttura radice. Se impostata su 0, il nodo della struttura è illimitato.
<code>stream-node-max-entries</code>	Tipo: INTEGER Valori consentiti: 0- Impostazione predefinita: 100	La struttura dati del flusso è una struttura radice che codifica più voci al suo interno. Utilizza questa configurazione per specificare il numero massimo di elementi che un singolo nodo può contenere prima di passare a un nuovo nodo durante l'accodamento di nuove voci di flusso. Se impostato su 0, il numero di elementi nel nodo dell'albero è illimitato.
<code>lazyfree-lazy- eviction</code>	Tipo: STRING Valori consentiti: sì, no Impostazione predefinita: no	Effettua un'eliminazione asincrona degli sfratti.

Nome	Informazioni	Descrizione
<code>active-defrag-ignore-bytes</code>	Tipo: INTEGER Valori consentiti: 1048576- Impostazione predefinita: 104857600	Quantità minima di scarto della frammentazione necessaria per avviare la deframmentazione attiva.
<code>lazyfree-lazy-expire</code>	Tipo: STRING Valori consentiti: sì, no Impostazione predefinita: no	Esegui un'eliminazione asincrona delle chiavi scadute.
<code>active-defrag-threshold-lower</code>	Tipo: INTEGER Valori consentiti: 1-100 Impostazione predefinita: 10	Percentuale minima di frammentazione necessaria per avviare la deframmentazione attiva.
<code>active-defrag-cycle-max</code>	Tipo: INTEGER Valori consentiti: 1-75 Impostazione predefinita: 75	Sforzo massimo per la deframmentazione in percentuale di CPU.
<code>lazyfree-lazy-server-del</code>	Tipo: STRING Valori consentiti: sì, no Impostazione predefinita: no	Esegue un'eliminazione asincrona per i comandi che aggiornano i valori.

Nome	Informazioni	Descrizione
<code>slowlog-log-slower-than</code>	Tipo: INTEGER Valori consentiti: 0- Impostazione predefinita: 10000	Il tempo di esecuzione massimo, in microsecondi, da superare affinché il comando venga registrato dalla funzionalità RedisSlow Log. Nota che un numero negativo disabilita lo slow log, mentre un valore pari a zero impone la registrazione di ogni comando.
<code>hash-max-ziplist-entries</code>	Tipo: INTEGER Valori consentiti: 0- Impostazione predefinita: 512	Determina la quantità di memoria utilizzata per gli hash. Gli hash con un numero di voci inferiore a quello specificato vengono archiviati con una codifica speciale che consente di risparmiare spazio.
<code>hash-max-ziplist-value</code>	Tipo: INTEGER Valori consentiti: 0- Impostazione predefinita: 64	Determina la quantità di memoria utilizzata per gli hash. Gli hash con voci di dimensioni inferiori al numero di byte specificato vengono archiviati con una codifica speciale che consente di risparmiare spazio.
<code>set-max-intset-entries</code>	Tipo: INTEGER Valori consentiti: 0- Impostazione predefinita: 512	Determina la quantità di memoria utilizzata per determinati tipi di set (stringhe di numeri interi in radice 10 nell'intervallo di interi con segno a 64 bit). Tali set con un numero di voci inferiore a quello specificato vengono archiviati con una codifica speciale che consente di risparmiare spazio.
<code>zset-max-ziplist-entries</code>	Tipo: INTEGER Valori consentiti: 0- Impostazione predefinita: 128	Determina la quantità di memoria utilizzata per i set ordinati. I set ordinati con un numero di elementi inferiore a quello specificato vengono archiviati con una codifica speciale che consente di risparmiare spazio.

Nome	Informazioni	Descrizione
<code>zset-max-ziplist-value</code>	Tipo: INTEGER Valori consentiti: 0- Impostazione predefinita: 64	Determina la quantità di memoria utilizzata per i set ordinati. I set ordinati con voci di dimensioni inferiori al numero di byte specificato vengono archiviati con una codifica speciale che consente di risparmiare spazio.
<code>tracking-table-max-keys</code>	Tipo: INTEGER Valori consentiti: 1-100000000 Impostazione predefinita: 1000000	Per aiutare il caching lato client, Redis supporta il monitoraggio di quali client hanno accesso a quali chiavi. Quando la chiave tracciata viene modificata, i messaggi di annullamento della convalida vengono inviati a tutti i client per notificare loro i valori memorizzati nella cache non sono più validi. Questo valore consente di specificare il limite superiore di questa tabella.
<code>acllog-max-len</code>	Tipo: INTEGER Valori consentiti: 1-10000 Impostazione predefinita: 128	Il numero massimo di voci nel registro ACL.

Nome	Informazioni	Descrizione
<code>active-expire-effort</code>	<p>Tipo: INTEGER</p> <p>Valori consentiti: 1-10</p> <p>Impostazione predefinita: 1</p>	<p>Redis elimina le chiavi che hanno superato la durata (TTL) con due meccanismi. In uno, si accede a una chiave e si trova scaduta. Nell'altro, un processo periodico campiona le chiavi e fa scadere quelle che hanno superato la loro durata (TTL). Questo parametro definisce la quantità di sforzo utilizzata da Redis per scadere gli elementi nel processo periodico.</p> <p>Il valore di default di 1 tenta di evitare di avere più del 10% delle chiavi scadute ancora in memoria. Inoltre cerca di evitare di consumare più del 25% della memoria totale e di aggiungere e latenza al sistema. È possibile aumentare questo valore fino a 10 per aumentare la quantità di sforzo speso per le chiavi in scadenza. Il compromesso è una CPU più alta e una latenza potenzialmente più elevata. Si consiglia un valore pari a 1, a meno che non si verifichi un utilizzo elevato della memoria e si possa tollerare un aumento dell'utilizzo della CPU.</p>
<code>lazyfree-lazy-user-del</code>	<p>Tipo: STRING</p> <p>Valori consentiti: sì, no</p> <p>Impostazione predefinita: no</p>	<p>Specifica se il comportamento predefinito del DEL comando agisce comeUNLINK.</p>
<code>activedefrag</code>	<p>Tipo: STRING</p> <p>Valori consentiti: sì, no</p> <p>Impostazione predefinita: no</p>	<p>Deframmentazione attiva della memoria abilitata.</p>

Nome	Informazioni	Descrizione
<code>maxclients</code>	Tipo: INTEGER Valori consentiti: 65000 Impostazione predefinita: 65000	Il numero massimo di client che possono essere connessi alla volta. Non modificabile
<code>client-query-buffer-limit</code>	Tipo: INTEGER Valori consentiti: 1048576-1073741824 Impostazione predefinita: 1073741824	Dimensione massima di un singolo buffer di query client. La modifica avviene immediatamente.
<code>proto-max-bulk-len</code>	Tipo: INTEGER Valori consentiti: 1048576-536870912 Impostazione predefinita: 536870912	Dimensione massima di una singola richiesta di elementi. La modifica avviene immediatamente.

Parametri specifici del tipo di nodo di MemoryDB

Sebbene la maggior parte dei parametri abbia un valore singolo, alcuni parametri hanno diversi valori in base al tipo di nodo utilizzato. La tabella seguente mostra il valore predefinito per ogni tipo di `maxmemory` nodo. Il valore di `maxmemory` è il numero massimo di byte disponibili sul nodo per utilizzo, dati e altro.

Tipo di nodo	Maxmemory
<code>db.r7g.large</code>	14037181030
<code>db.r7g.xlarge</code>	28261849702

Tipo di nodo	Maxmemory
db.r7g.2xlarge	56711183565
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

Tutti i tipi di istanze MemoryDB devono essere creati in un VPC Amazon Virtual Private Cloud.

Tutorial: Configurazione di una funzione Lambda per accedere a MemoryDB in un Amazon VPC

In questo tutorial puoi imparare a:

- Crea un cluster MemoryDB nel tuo Amazon Virtual Private Cloud (Amazon VPC) predefinito nella regione us-east-1.
- Crea una funzione Lambda per accedere al cluster. Quando si crea la funzione Lambda, si forniscono gli ID della sottorete nell'Amazon VPC e un gruppo di sicurezza VPC per consentire alla funzione Lambda di accedere alle risorse nel VPC. A titolo illustrativo in questo tutorial, la funzione Lambda genera un UUID, lo scrive nel cluster e lo recupera dal cluster.
- Richiama la funzione Lambda manualmente e verifica che abbia avuto accesso al cluster nel tuo VPC.
- Pulisci la funzione Lambda, il cluster e il ruolo IAM configurati per questo tutorial.

Argomenti

- [Fase 1: creazione di un cluster](#)
- [Passaggio 2: creazione di una funzione Lambda](#)
- [Fase 3: esecuzione del test della funzione Lambda](#)
- [Fase 4: Pulizia \(opzionale\)](#)

Fase 1: creazione di un cluster

Per creare un cluster, segui questi passaggi.

Argomenti

- [Fase 1.1: Creare un cluster](#)
- [Passaggio 1.2: Copia l'endpoint del cluster](#)

- [Fase 1.3: Creazione del ruolo IAM](#)
- [Passaggio 1.4: Creare una lista di controllo degli accessi \(ACL\)](#)

Fase 1.1: Creare un cluster

In questa fase, crei un cluster nel VPC Amazon predefinito nella regione us-east-1 del tuo account utilizzando (CLI). AWS Command Line Interface Per informazioni sulla creazione di cluster utilizzando la console o l'API di MemoryDB, consulta. [Fase 1: creazione di un cluster](#)

```
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
```

Il valore del campo Stato è impostato su CREATING. MemoryDB può impiegare alcuni minuti per completare la creazione del cluster.

Passaggio 1.2: Copia l'endpoint del cluster

Verifica che MemoryDB abbia terminato la creazione del cluster con il comando. `describe-clusters`

```
aws memorydb describe-clusters \
--cluster-name cluster-01
```

Copia l'indirizzo dell'endpoint del cluster mostrato nell'output. Avrai bisogno di questo indirizzo quando crei il pacchetto di implementazione per la funzione Lambda.

Fase 1.3: Creazione del ruolo IAM

1. Crea un documento della policy di attendibilità IAM per il ruolo, come mostrato di seguito, che consenta all'account di assumere il nuovo ruolo. Salva la policy in un file denominato `trust-policy.json`. Assicurati di sostituire `account_id 123456789012` in questa politica con il tuo `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. Crea un documento della policy IAM, come mostrato di seguito. Salva la policy in un file denominato `policy.json`. Assicurati di sostituire `account_id` 123456789012 in questa politica con il tuo `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. Crea un ruolo IAM.

```

aws iam create-role \
--role-name "memorydb-iam-auth-app" \
--assume-role-policy-document file://trust-policy.json

```

4. Creare la policy IAM.

```

aws iam create-policy \
--policy-name "memorydb-allow-all" \

```

```
--policy-document file://policy.json
```

5. Allegare la policy IAM al ruolo. Assicurati di sostituire `account_id 123456789012` in questo `policy-arn` con il tuo `account_id`.

```
aws iam attach-role-policy \  
  --role-name "memorydb-iam-auth-app" \  
  --policy-arn "arn:aws:iam::123456789012:policy/memorydb-allow-all"
```

Passaggio 1.4: Creare una lista di controllo degli accessi (ACL)

1. Crea un nuovo utente attivato da IAM.

```
aws memorydb create-user \  
  --user-name iam-user-01 \  
  --authentication-mode Type=iam \  
  --access-string "on ~* +@all"
```

2. Crea un ACL e collegalo al cluster.

```
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
```

Passaggio 2: creazione di una funzione Lambda

Per creare una funzione Lambda, procedi nel seguente modo.

Argomenti

- [Fase 2.1: creazione del pacchetto di implementazione](#)
- [Fase 2.2: creazione del ruolo IAM \(ruolo di esecuzione\)](#)
- [Fase 2.3: caricamento del pacchetto di implementazione \(creazione della funzione Lambda\)](#)

Fase 2.1: creazione del pacchetto di implementazione

In questo tutorial, forniamo codice di esempio in Python per la tua funzione Lambda.

Python

L'esempio seguente di codice Python legge e scrive un elemento nel cluster MemoryDB. Copia il codice e salvalo in un file denominato `app.py`. Assicurati di sostituire il `cluster_endpoint` valore nel codice con l'indirizzo dell'endpoint che hai copiato nel passaggio 1.2.

```
from typing import Tuple, Union
from urllib.parse import ParseResult, urlencode, urlunparse

import boto3.session
import redis
from boto3.model import ServiceId
from boto3.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 = boto3.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"

```

Questo codice utilizza la `redis-py` libreria Python per inserire elementi nel cluster e recuperarli. Questo codice viene utilizzato `cachetools` per memorizzare nella cache i token di autenticazione IAM generati per 15 minuti. Per creare un pacchetto di distribuzione contenente `redis-py` e `cachetools`, procedi nel seguente modo.

Nella directory del progetto contenente il file del codice `app.py` sorgente, create un pacchetto di cartelle in cui installare le `cachetools` librerie `redis-py` and.

```
mkdir package
```

Installa `redis-py` e `cachetools` usa `pip`.

```
pip install --target ./package redis
pip install --target ./package cachetools
```

Crea un file `.zip` contenente le librerie `redis-py` and `cachetools`. In Linux e macOS, esegui il seguente comando. In Windows, utilizzate l'utilità `zip` preferita per creare un file `.zip` con le `cachetools` librerie `redis-py` and alla radice.

```
cd package
zip -r ../my_deployment_package.zip
```

Aggiungi il codice della funzione al file `.zip`. Su Linux e macOS, esegui il comando seguente. In Windows, utilizzate l'utilità `zip` preferita per aggiungere `app.py` alla radice del file con estensione `zip`.

```
cd ..
zip my_deployment_package.zip app.py
```

Fase 2.2: creazione del ruolo IAM (ruolo di esecuzione)

Allega la policy AWS gestita denominata `AWSLambdaVPCLambdaAccessExecutionRole` al ruolo.

```
aws iam attach-role-policy \
  --role-name "memorydb-iam-auth-app" \
  --policy-arn "arn:aws:iam::aws:policy/service-role/AWSLambdaVPCLambdaAccessExecutionRole"
```

Fase 2.3: caricamento del pacchetto di implementazione (creazione della funzione Lambda)

In questo passaggio, si crea la funzione Lambda (AccessMemoryDB) utilizzando il comando AWS CLI `create-function`.

Dalla directory del progetto che contiene il file.zip del pacchetto di distribuzione, esegui il seguente comando Lambda `create-function` CLI.

Per l'opzione `role`, utilizzate l'ARN del ruolo di esecuzione creato nel passaggio 2.2. Per `vpc-config` inserisci gli elenchi separati da virgole delle sottoreti del tuo VPC predefinito e dell'ID del gruppo di sicurezza del tuo VPC predefinito. Questi valori sono disponibili nella console Amazon VPC. Per trovare le sottoreti del tuo VPC predefinito, scegli i tuoi VPC, quindi scegli il VPC predefinito del tuo account AWS. Per trovare il gruppo di sicurezza per questo VPC, vai su Sicurezza e scegli Gruppi di sicurezza. Assicurati di aver selezionato la regione `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
```

Fase 3: esecuzione del test della funzione Lambda

In questo passaggio, si richiama la funzione Lambda manualmente utilizzando il comando `invoke`. Quando la funzione Lambda viene eseguita, genera un UUID e lo scrive nella ElastiCache cache specificata nel codice Lambda. Successivamente la funzione Lambda recupera la voce dalla cache.

1. Invoca la funzione Lambda AccessMemory (DB) utilizzando AWS Lambda il comando `invoke`.

```
aws lambda invoke \  
--function-name AccessMemoryDB \  
--region us-east-1 \  
output.txt
```

2. Eseguire le operazioni seguenti per verificare che la funzione Lambda sia stata eseguita nel modo corretto:
 - Esaminare il file output.txt.
 - Verifica i risultati in CloudWatch Logs aprendo la CloudWatch console e scegliendo il gruppo di log per la tua funzione (/aws/lambda/). AccessRedis Il flusso di log genera un output simile al seguente:

```
Success: Inserted 826e70c5f4d2478c8c18027125a3e01e. Fetched
826e70c5f4d2478c8c18027125a3e01e from MemoryDB.
```

- Controlla i risultati nella console. AWS Lambda

Fase 4: Pulizia (opzionale)

Per eseguire la pulizia, procedi nel seguente modo.

Argomenti

- [Fase 4.1: Eliminare la funzione Lambda](#)
- [Fase 4.2: Eliminare il cluster MemoryDB](#)
- [Fase 4.3: Rimuovere il ruolo e le politiche IAM](#)

Fase 4.1: Eliminare la funzione Lambda

```
aws lambda delete-function \  
--function-name AccessMemoryDB
```

Fase 4.2: Eliminare il cluster MemoryDB

Elimina il cluster.

```
aws memorydb delete-cluster \  
--cluster-name cluster-01
```

Rimuovi utente e ACL.

```
aws memorydb delete-user \  

```

```
--user-id iam-user-01

aws memorydb delete-acl \  
--acl-name iam-acl-01
```

Fase 4.3: Rimuovere il ruolo e le politiche 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/AWSLambdaVPCLambdaAccessExecutionRole"

aws iam delete-role \  
--role-name "memorydb-iam-auth-app"

aws iam delete-policy \  
--policy-arn "arn:aws:iam::123456789012:policy/memorydb-allow-all"
```

Ricerca vettoriale

Questa funzionalità è disponibile in anteprima per MemoryDB for Redis ed è soggetta a modifiche.

La ricerca vettoriale per MemoryDB estende le funzionalità di MemoryDB. La ricerca vettoriale può essere utilizzata insieme alle funzionalità MemoryDB esistenti. Le applicazioni che non utilizzano la ricerca vettoriale non sono influenzate dalla sua presenza. L'anteprima della ricerca vettoriale è disponibile dalla versione MemoryDB 7.1 in poi nelle seguenti regioni: Stati Uniti orientali (Virginia settentrionale e Ohio), Stati Uniti occidentali (Oregon), UE (Irlanda) e Asia Pacifico (Tokyo).

La ricerca vettoriale per Amazon MemoryDB for Redis semplifica l'architettura dell'applicazione offrendo al contempo una ricerca vettoriale ad alta velocità. La ricerca vettoriale per MemoryDB è ideale per i casi d'uso in cui le massime prestazioni e la scalabilità sono i criteri di selezione più importanti. Puoi utilizzare i dati di MemoryDB esistenti o l'API Redis per creare casi d'uso di machine learning e intelligenza artificiale generativa, come la generazione potenziata dal recupero, il rilevamento delle anomalie, il recupero di documenti e i consigli in tempo reale.

Argomenti

- [Panoramica della ricerca vettoriale](#)
- [Caratteristiche e limiti della ricerca vettoriale](#)
- [Casi d'uso](#)
- [Utilizzando il AWS Management Console](#)
- [Utilizzando il AWS Command Line Interface](#)
- [Comandi di ricerca vettoriale](#)

Panoramica della ricerca vettoriale

Questa funzionalità è disponibile in anteprima per MemoryDB for Redis ed è soggetta a modifiche.

La ricerca vettoriale si basa sulla creazione, la manutenzione e l'uso di indici. Ogni operazione di ricerca vettoriale specifica un singolo indice e il suo funzionamento è limitato a quell'indice, ovvero le operazioni su un indice non sono influenzate dalle operazioni su nessun altro indice. Ad eccezione

delle operazioni di creazione e distruzione degli indici, è possibile eseguire un numero qualsiasi di operazioni su qualsiasi indice in qualsiasi momento, il che significa che a livello di cluster è possibile eseguire più operazioni su più indici contemporaneamente.

I singoli indici sono oggetti denominati che esistono in uno spazio dei nomi univoco, separato dagli altri namespace Redis: chiavi, funzioni, ecc. Ogni indice è concettualmente simile a una tabella di database convenzionale in quanto è strutturato in due dimensioni: colonna e righe. Ogni riga della tabella corrisponde a una chiave Redis. Ogni colonna dell'indice corrisponde a un membro o parte di quella chiave. All'interno di questo documento i termini chiave, riga e record sono identici e utilizzati in modo intercambiabile. Allo stesso modo, i termini colonna, campo, percorso e membro sono essenzialmente identici e sono anche usati in modo intercambiabile.

Non esistono comandi speciali per aggiungere, eliminare o modificare dati indicizzati. Piuttosto, anche JSON i comandi esistenti HASH o che modificano una chiave presente in un indice aggiornano automaticamente l'indice.

Argomenti

- [Indici e keyspace Redis](#)
- [Tipi di campi dell'indice](#)
- [Algoritmi di indice vettoriale](#)
- [Espressione di interrogazione di ricerca vettoriale](#)
- [Comando INFO](#)
- [Sicurezza della ricerca vettoriale](#)

Indici e keyspace Redis

Gli indici sono costruiti e gestiti su un sottoinsieme dello spazio chiave Redis. Più indici possono scegliere sottoinsiemi disgiunti o sovrapposti dello spazio chiave Redis senza limitazioni. Lo spazio chiave per ogni indice è definito da un elenco di prefissi chiave forniti al momento della creazione dell'indice. L'elenco dei prefissi è facoltativo e, se omesso, l'intero keyspace Redis farà parte di quell'indice. Gli indici vengono inoltre digitati in quanto coprono solo le chiavi che hanno un tipo corrispondente. Attualmente sono supportati solo gli indici JSON e HASH. Un indice HASH indicizza solo le chiavi HASH incluse nel relativo elenco di prefissi e analogamente un indice JSON indicizza solo le chiavi JSON incluse nel relativo elenco di prefissi. Le chiavi all'interno dell'elenco dei prefissi dello spazio dei tasti di un indice che non hanno il tipo designato vengono ignorate e non influiscono sulle operazioni di ricerca.

Quando un comando HASH o JSON modifica una chiave che si trova all'interno di uno spazio chiave di un indice, tale indice viene aggiornato. Questo processo prevede l'estrazione dei campi dichiarati per ogni indice e l'aggiornamento dell'indice con il nuovo valore. Il processo di aggiornamento viene eseguito in un thread in background, il che significa che gli indici sono coerenti solo alla fine con il contenuto del loro keyspace. Pertanto l'inserimento o l'aggiornamento di una chiave non sarà visibile nei risultati di ricerca per un breve periodo di tempo. Durante i periodi di intenso carico del sistema e/o di forte mutazione dei dati, il ritardo di visibilità può aumentare.

La creazione di un indice è un processo in più fasi. Il primo passo consiste nell'eseguire il comando [FT.CREATE](#) che definisce l'indice. L'esecuzione corretta di una creazione avvia automaticamente il secondo passaggio: il backfilling. Il processo di riempimento viene eseguito in un thread in background e analizza lo spazio delle chiavi Redis alla ricerca di chiavi che si trovano all'interno dell'elenco di prefissi del nuovo indice. Ogni chiave trovata viene aggiunta all'indice. Alla fine viene scansionato l'intero keyspace, completando il processo di creazione dell'indice. Nota che mentre il processo di riempimento dell'indice è in esecuzione, le mutazioni delle chiavi indicizzate sono consentite, non ci sono restrizioni e il processo di riempimento dell'indice non verrà completato finché tutte le chiavi non saranno indicizzate correttamente. Le operazioni di interrogazione tentate mentre un indice è in fase di riempimento non sono consentite e vengono terminate con un errore. Il completamento del processo di riempimento può essere determinato dall'output del `FT.INFO` comando per quell'indice ('backfill_status').

Tipi di campi dell'indice

Ogni campo (colonna) di un indice ha un tipo specifico che viene dichiarato al momento della creazione dell'indice e una posizione all'interno di una chiave. Per le chiavi HASH, la posizione è il nome del campo all'interno dell'HASH. Per le chiavi JSON, la posizione è una descrizione del percorso JSON. Quando una chiave viene modificata, i dati associati ai campi dichiarati vengono estratti, convertiti nel tipo dichiarato e memorizzati nell'indice. Se i dati mancano o non possono essere convertiti correttamente nel tipo dichiarato, quel campo viene ommesso dall'indice. Esistono quattro tipi di campi, come spiegato di seguito:

- I campi numerici contengono un solo numero. Per i campi JSON, è necessario seguire le regole numeriche dei numeri JSON. Per HASH, il campo dovrebbe contenere il testo ASCII di un numero scritto nel formato standard per numeri a virgola fissa o mobile. Indipendentemente dalla rappresentazione all'interno della chiave, questo campo viene convertito in un numero a virgola mobile a 64 bit per la memorizzazione all'interno dell'indice. I campi numerici possono essere utilizzati con l'operatore di ricerca per intervalli. Poiché i numeri sottostanti sono memorizzati in

virgola mobile con i relativi limiti di precisione, si applicano le normali regole sui confronti numerici per i numeri in virgola mobile.

- I campi tag contengono zero o più valori di tag codificati come una singola stringa UTF-8. La stringa viene analizzata in valori di tag utilizzando un carattere separatore (l'impostazione predefinita è una virgola ma può essere sovrascritta) con gli spazi bianchi iniziali e finali rimossi. Qualsiasi numero di valori di tag può essere contenuto in un singolo campo di tag. I campi tag possono essere utilizzati per filtrare le query per l'equivalenza dei valori dei tag con un confronto con o senza distinzione tra maiuscole e minuscole.
- I campi di testo contengono una serie di byte che non devono necessariamente essere conformi a UTF-8. I campi di testo possono essere utilizzati per decorare i risultati delle query con valori significativi per l'applicazione. Ad esempio un URL o il contenuto di un documento, ecc.
- I campi vettoriali contengono un vettore di numeri noto anche come incorporamento. I campi vettoriali supportano la ricerca K-Nearest Neighbor Searching (KNN) di vettori di dimensioni fisse utilizzando un algoritmo e una metrica di distanza specificati. Per gli indici HASH, il campo deve contenere l'intero vettore codificato in formato binario (Little-endian IEEE 754). Per le chiavi JSON, il percorso deve fare riferimento a un array della dimensione corretta pieno di numeri. Nota che quando un array JSON viene utilizzato come campo vettoriale, la rappresentazione interna dell'array all'interno della chiave JSON viene convertita nel formato richiesto dall'algoritmo selezionato, riducendo il consumo di memoria e la precisione. Le successive operazioni di lettura che utilizzano i comandi JSON produrranno un valore di precisione ridotto.

Algoritmi di indice vettoriale

Sono disponibili due algoritmi di indice vettoriale:

- Flat — L'algoritmo Flat è un'elaborazione lineare a forza bruta di ogni vettore dell'indice, che fornisce risposte esatte entro i limiti della precisione dei calcoli della distanza. Grazie all'elaborazione lineare dell'indice, i tempi di esecuzione di questo algoritmo possono essere molto elevati per indici di grandi dimensioni.
- HNSW (Hierarchical Navigable Small Worlds) — L'algoritmo HNSW è un'alternativa che fornisce un'approssimazione della risposta corretta in cambio di tempi di esecuzione notevolmente inferiori. L'algoritmo è controllato da tre parametri e. `M` `EF_CONSTRUCTION` `EF_RUNTIME` I primi due parametri vengono specificati al momento della creazione dell'indice e non possono essere modificati. Il `EF_RUNTIME` parametro ha un valore predefinito che viene specificato al momento della creazione dell'indice, ma può essere sovrascritto in ogni singola operazione di interrogazione in seguito. Questi tre parametri interagiscono per bilanciare il consumo di memoria e CPU durante

le operazioni di inserimento e interrogazione, nonché per controllare la qualità dell'approssimazione di una ricerca KNN esatta (nota come rapporto di richiamo).

Entrambi gli algoritmi di ricerca vettoriale (Flat e HNSW) supportano un parametro opzionale. `INITIAL_CAP` Quando specificato, questo parametro prealloca la memoria per gli indici, con conseguente riduzione del sovraccarico di gestione della memoria e aumento delle velocità di ingestione vettoriale.

Gli algoritmi di ricerca vettoriale come HNSW potrebbero non gestire in modo efficiente l'eliminazione o la sovrascrittura dei vettori precedentemente inseriti. L'uso di queste operazioni può comportare un consumo eccessivo di memoria indicizzata e/o una riduzione della qualità dei richiami. La reindicizzazione è un metodo per ripristinare l'utilizzo ottimale della memoria e/o il richiamo.

Espressione di interrogazione di ricerca vettoriale

I comandi [FT.SEARCH](#) e [FT.AGGREGATE](#) richiedono un'espressione di interrogazione. Questa espressione è un parametro a stringa singola composto da uno o più operatori. Ogni operatore utilizza un campo dell'indice per identificare un sottoinsieme delle chiavi dell'indice. È possibile combinare più operatori utilizzando combinatori booleani e parentesi per migliorare o limitare ulteriormente il set di chiavi (o set di risultati) raccolto.

Carattere jolly

L'operatore jolly, l'asterisco (*), corrisponde a tutte le chiavi dell'indice.

Intervallo numerico

L'operatore di intervallo numerico ha la seguente sintassi:

```
<range-search> ::= '@' <numeric-field-name> ':' '[' <bound> <bound> ']'  
<bound> ::= <number> | '(' <number>  
<number> ::= <integer> | <fixed-point> | <floating-point> | 'Inf' | '-Inf' | '+Inf'
```

< numeric-field-name > deve essere un campo di tipo dichiarato. `NUMERIC` Per impostazione predefinita, il limite è inclusivo, ma è possibile utilizzare una parentesi aperta iniziale '[' per rendere esclusivo un limite. La ricerca per intervallo può essere convertita in un unico confronto relazionale (<, <=, >, >=) utilizzando `Inf` `+Inf` o `-Inf` come uno dei limiti. Indipendentemente dal formato numerico specificato (intero, a virgola fissa, a virgola mobile, infinito), il numero viene convertito in virgola mobile a 64 bit per eseguire confronti, riducendo di conseguenza la precisione.

Example Esempi

```
@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
```

Confronta tag

L'operatore di confronto dei tag ha la seguente sintassi:

```
<tag-search> ::= '@' <tag-field-name> ':' '{' <tag> [ '|' <tag> ]* '}'
```

Se uno qualsiasi dei tag nell'operatore corrisponde a uno dei tag nel campo dei tag del record, il record viene incluso nel set di risultati. Il campo progettato da <tag-field-name> deve essere un campo dell'indice dichiarato con type. TAG Esempi di confronto tra tag sono:

```
@tag-field:{ atag }
@tag-field: { tag1 | tag2 }
```

Combinazioni booleane

I set di risultati di un operatore numerico o di tag possono essere combinati utilizzando la logica booleana: e/or. Le parentesi possono essere utilizzate per raggruppare gli operatori e/o modificare l'ordine di valutazione. La sintassi degli operatori logici booleani è:

```
<expression> ::= <phrase> | <phrase> '|' <expression> | '(' <expression> ')'
<phrase> ::= <term> | <term> <phrase>
<term> ::= <range-search> | <tag-search> | '*'
```

Più termini combinati in una frase sono «and» -ed. Le frasi multiple combinate con la pipe (|) sono «or» -ed.

Ricerca vettoriale

L'operatore di ricerca vettoriale esegue una ricerca con K punti vicini all'interno di un indice di campi vettoriali. La sintassi di una ricerca vettoriale è:

```
<vector-search> ::= <expression> '=>[KNN' <k> '@' <vector-field-name> '$' <parameter-name> <modifiers> ']'
<modifiers> ::= [ 'EF_RUNTIME' <integer> ] [ 'AS' <distance-field-name>]
```

La ricerca vettoriale viene applicata solo ai vettori che soddisfano i <expression> quali possono essere una qualsiasi combinazione degli operatori sopra definiti: wildcard, range search, tag search e/o relative combinazioni booleane.

- <k> è un numero intero che specifica il numero di vettori vicini più prossimi da restituire.
- <vector-field-name> deve specificare un campo di tipo dichiarato. VECTOR
- <parameter-name> field specifica una delle voci della PARAM tabella del FT.AGGREGATE comando FT.SEARCH or. Questo parametro è il valore vettoriale di riferimento per il calcolo della distanza. Il valore del vettore è codificato nel PARAM valore in formato binario IEEE 754 di little-endian (stessa codifica utilizzata per un campo vettoriale HASH)
- Per gli indici vettoriali di tipo HNSW, la EF_RUNTIME clausola opzionale può essere utilizzata per sovrascrivere il valore predefinito del parametro stabilito al momento della creazione dell'indice. EF_RUNTIME
- L'opzione <distance-field-name> fornisce un nome di campo per il set di risultati che contiene la distanza calcolata tra il vettore di riferimento e la chiave individuata.

Comando INFO

La ricerca vettoriale amplia il comando Redis [INFO](#) con diverse sezioni aggiuntive di statistiche e contatori. Una richiesta di recupero della sezione SEARCH recupererà tutte le seguenti sezioni:

Sezione **search_memory**

Nome	Descrizione
search_used_memory_bytes	Numero di byte di memoria consumati in tutte le strutture di dati di ricerca
search_used_memory_human	Versione leggibile dall'uomo di cui sopra

Sezione `search_index_stats`

Nome	Descrizione
<code>numero_di_indici_di_ricerca</code>	Numero di indici creati
<code>search_num_fulltext_indexes</code>	Numero di campi non vettoriali in tutti gli indici
<code>search_num_vector_indexes</code>	Numero di campi vettoriali in tutti gli indici
<code>search_num_hash_indexes</code>	Numero di indici su chiavi di tipo HASH
<code>search_num_json_indexes</code>	Numero di indici su chiavi di tipo JSON
<code>search_total_indexed_keys</code>	Numero totale di chiavi in tutti gli indici
<code>search_total_indexed_vectors</code>	Numero totale di vettori in tutti gli indici
<code>search_total_indexed_hash_keys</code>	Numero totale di chiavi di tipo HASH in tutti gli indici
<code>search_total_indexed_json_keys</code>	Numero totale di chiavi di tipo JSON in tutti gli indici
<code>search_total_index_size</code>	Byte utilizzati da tutti gli indici
<code>search_total_fulltext_index_size</code>	Byte utilizzati da strutture indicizzate non vettoriali
<code>search_total_vector_index_size</code>	Byte utilizzati dalle strutture degli indici vettoriali
<code>search_max_index_lag_ms</code>	Ritardo di inserimento durante l'ultimo aggiornamento del batch di importazione

Sezione **search_ingestion**

Nome	Descrizione
search_background_indexing_status	Stato di ingestione. NO_ACTIVITY significa inattivo. Altri valori indicano che ci sono chiavi in fase di ingestione.
search_ingestion_paused	Tranne durante il riavvio, questo dovrebbe sempre essere «no».

Sezione **search_backfill**

Note

Alcuni dei campi documentati in questa sezione sono visibili solo quando è attualmente in corso un riempimento.

Nome	Descrizione
search_num_active_backfills	Numero di attività di riempimento correnti
search_backfills_paused	Tranne quando la memoria è esaurita, dovrebbe sempre essere «no».
search_highest_backfill_progress_percentage	% di completamento (0-100) del riempimento più completato
search_lowest_backfill_progress_percentage	% di completamento (0-100) del riempimento meno completato

Sezione `search_query`

Nome	Descrizione
<code>search_num_active_queries</code>	Numero di comandi and attualmente in corso FT.SEARCH FT.AGGREGATE

Sicurezza della ricerca vettoriale

I meccanismi di sicurezza [Redis ACL \(Access Control Lists\)](#) per l'accesso ai comandi e ai dati sono estesi per controllare la funzione di ricerca. Il controllo ACL dei singoli comandi di ricerca è completamente supportato. Viene fornita una nuova categoria ACL e molte delle categorie esistenti (`@fast`, `@read@write`, ecc.) vengono aggiornate per includere i nuovi comandi. `@search` I comandi di ricerca non modificano i dati chiave, il che significa che il meccanismo ACL esistente per l'accesso in scrittura viene preservato. Le regole di accesso per le operazioni HASH e JSON non vengono modificate dalla presenza di un indice; a tali comandi viene comunque applicato il normale controllo dell'accesso a livello di chiave.

L'accesso ai comandi di ricerca con un indice è inoltre controllato tramite Redis ACL. I controlli di accesso vengono eseguiti a livello dell'intero indice, non a livello di chiave. Ciò significa che l'accesso a un indice viene concesso a un utente solo se tale utente è autorizzato ad accedere a tutte le chiavi possibili all'interno dell'elenco dei prefissi dello spazio chiave di quell'indice. In altre parole, il contenuto effettivo di un indice non controlla l'accesso. Piuttosto, sono i contenuti teorici di un indice, come definito dall'elenco dei prefissi, che viene utilizzato per il controllo di sicurezza. Può essere facile creare una situazione in cui un utente abbia accesso in lettura e/o scrittura a una chiave ma non sia in grado di accedere a un indice contenente quella chiave. Tieni presente che per creare o utilizzare un indice è necessario solo l'accesso in lettura allo spazio delle chiavi: la presenza o l'assenza di accesso in scrittura non viene considerata.

Per ulteriori informazioni sull'utilizzo degli ACL con MemoryDB, consultate [Authenticating users with Access Control Lists \(ACL\)](#).

Caratteristiche e limiti della ricerca vettoriale

Questa funzionalità è disponibile in anteprima per MemoryDB for Redis ed è soggetta a modifiche.

Disponibilità della ricerca vettoriale

La configurazione MemoryDB abilitata alla ricerca vettoriale è supportata sui tipi di nodi R6g, R7g e T4g ed è disponibile nelle seguenti AWS regioni: Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (Oregon), Asia Pacifico (Tokyo) ed Europa (Irlanda).

Restrizioni parametriche

La tabella seguente mostra i limiti per vari elementi di ricerca vettoriale nell'anteprima:

Elemento	Valore massimo
Numero di dimensioni in un vettore	32768
Numero di indici che possono essere creati	10
Numero di campi in un indice	50
Numero di operazioni di riempimento simultane e dell'indice FT.CREATE	1
Clausola FT.SEARCH e FT.AGGREGATE TIMEOUT (millisecondi)	60000
Numero di fasi della pipeline nel comando FT.AGGREGATE	32
Numero di campi nella clausola FT.AGGREGATE LOAD	1.024
Numero di campi nella clausola FT.AGGREGATE GROUPBY	16
Numero di campi nella clausola FT.AGGREGATE SORTBY	16
Numero di parametri nella clausola FT.AGGREGATE PARAM	32
Parametro HNSW M	512

Elemento	Valore massimo
Parametro HNSW EF_CONSTRUCTION	4096
Parametro HNSW EF_RUNTIME	4096

Limiti di scalabilità

La ricerca vettoriale per MemoryDB è attualmente limitata a un singolo frammento e il ridimensionamento orizzontale non è supportato. La ricerca vettoriale supporta il ridimensionamento verticale e di replica.

Restrizioni operative

Persistenza e riempimento dell'indice

L'anteprima della ricerca vettoriale mantiene la definizione degli indici, ma non il loro contenuto. Pertanto, qualsiasi richiesta o evento operativo che causa l'avvio o il riavvio di un nodo richiede la ricostruzione di tutti gli indici sulla base della definizione e dei dati chiave di origine. Il processo di ricostruzione viene avviato automaticamente una volta ripristinati tutti i dati: non è richiesta alcuna azione da parte dell'utente per avviarlo. La ricostruzione viene eseguita come operazione di riempimento non appena i dati vengono ripristinati. Dal punto di vista funzionale, ciò equivale all'esecuzione automatica da parte del sistema di un comando [FT.CREATE per ogni indice](#) definito. Si noti che il nodo diventa disponibile per le operazioni dell'applicazione non appena i dati vengono ripristinati, ma probabilmente prima del completamento del riempimento dell'indice, il che significa che i backfill diventeranno nuovamente visibili alle applicazioni; ad esempio, i comandi di ricerca che utilizzano gli indici di riempimento potrebbero essere rifiutati. Per ulteriori informazioni sul backfilling, vedere [Panoramica della ricerca vettoriale](#)

Il completamento del riempimento dell'indice non è sincronizzato tra un primario e una replica. Questa mancanza di sincronizzazione può diventare inaspettatamente visibile alle applicazioni, pertanto è consigliabile che le applicazioni verifichino il completamento del backfill sui file primari e su tutte le repliche prima di iniziare le operazioni di ricerca.

Importazione/esportazione di istantanee e Live Migration

La presenza di indici di ricerca in un file RDB limita la trasportabilità compatibile di tali dati. Il formato degli indici definito dall'edizione di anteprima è compreso solo da un altro cluster di edizioni di

anteprima. Pertanto, i file RDB con indici di ricerca possono essere trasferiti o utilizzati solo da cluster MemoryDB abilitati all'anteprima.

Tuttavia, i file RDB che non contengono indici non sono soggetti a restrizioni in questo modo. Pertanto i dati all'interno di un cluster di anteprima possono essere esportati in cluster non di anteprima eliminando gli indici prima dell'esportazione.

Consumo di memoria

L'attuale implementazione degli indici vettoriali consuma circa il doppio della quantità di memoria utilizzata dall'implementazione General Availability.

Memoria insufficiente durante il riempimento

Analogamente alle operazioni di scrittura di Redis, un riempimento dell'indice è soggetto a limitazioni. out-of-memory Se la memoria Redis è piena mentre è in corso un riempimento, tutti i backfill vengono messi in pausa. Se la memoria diventa disponibile, il processo di riempimento viene ripreso. È anche possibile eliminare e indicizzare quando il riempimento è in pausa a causa dell'esaurimento della memoria.

Transazioni

I comandi `FT.CREATE`, `FT.DROPINDEX`, `FT.ALIASADDFT`, `FT.ALIASDEL`, e `FT.ALIASUPDATE` non possono essere eseguiti in un contesto transazionale, cioè non all'interno di un blocco `MULTI/EXEC` o all'interno di uno script `LUA` o `FUNCTION`.

Casi d'uso

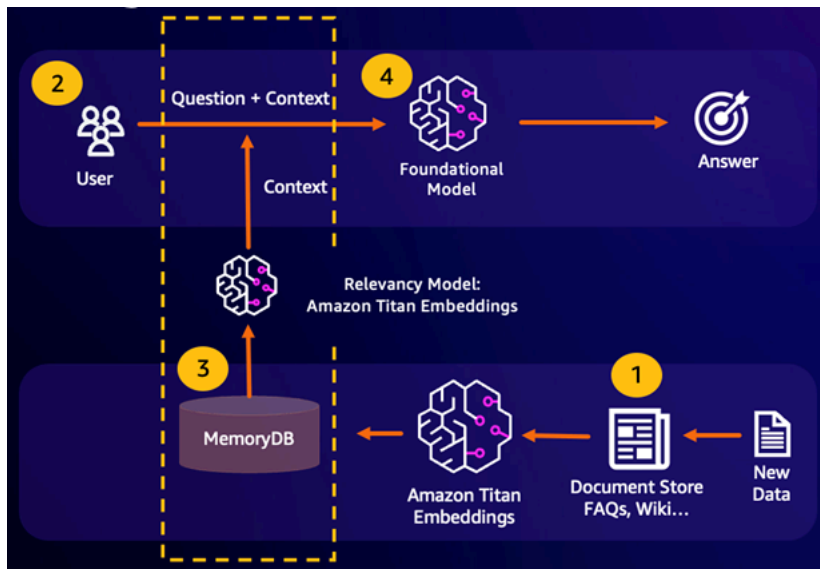
Questa funzionalità è disponibile in anteprima per MemoryDB for Redis ed è soggetta a modifiche.

Di seguito sono riportati i casi d'uso della ricerca vettoriale.

Retrieval Augmented Generation (RAG)

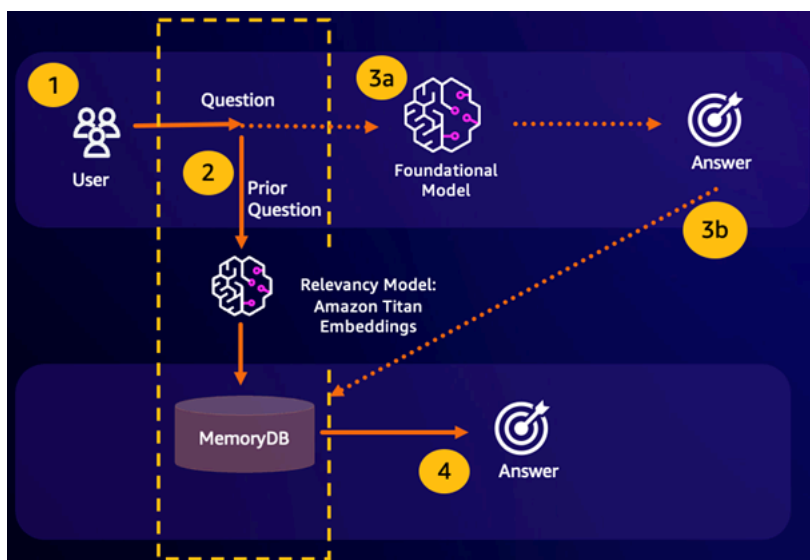
Retrieval Augmented Generation (RAG) sfrutta la ricerca vettoriale per recuperare i passaggi pertinenti da un ampio corpus di dati per ampliare un ampio modello linguistico (LLM). In particolare, un codificatore incorpora il contesto di input e la query di ricerca in vettori, quindi utilizza la ricerca approssimativa del vicino più prossimo per trovare passaggi semanticamente simili. Questi passaggi

recuperati vengono concatenati con il contesto originale per fornire ulteriori informazioni pertinenti all'LLM e restituire una risposta più accurata all'utente.



Memoria buffer Foundation Model (FM)

La memoria buffer del modello Foundation (FM) è un processo per ridurre i costi di calcolo memorizzando i risultati precedenti dell'FM. Riutilizzando i risultati precedenti delle inferenze precedenti anziché ricalcolarli, la memoria buffer FM riduce la quantità di calcolo richiesta durante l'inferenza tramite FM. Questa memoria buffer FM consente ai modelli linguistici di grandi dimensioni di rispondere più rapidamente con costi inferiori grazie ai costi di servizio dell'FM.



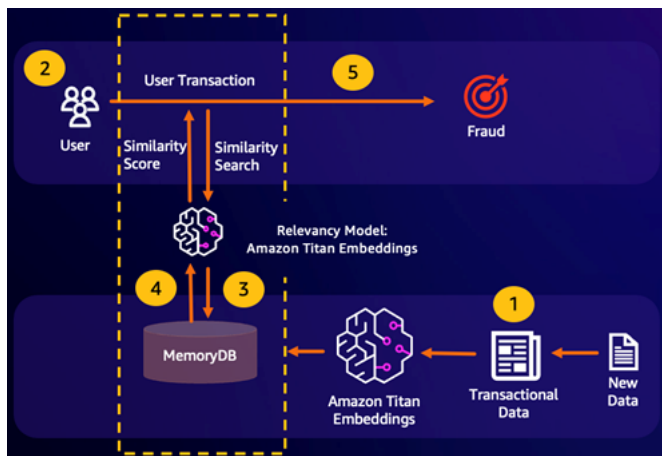
- Risultato della ricerca semantica: se la query di un cliente è semanticamente simile a una domanda precedente sulla base di un punteggio di somiglianza definito, la memoria buffer FM (MemoryDB)

restituirà la risposta alla domanda precedente nel passaggio 4 e non richiamerà l'FM durante i passaggi 3. In questo modo si eviteranno la latenza del modello di base (FM) e i costi sostenuti, garantendo un'esperienza più rapida per il cliente.

- Ricerca semantica non riuscita: se la query di un cliente non è semanticamente simile in base a un punteggio di somiglianza definito a una query precedente, un cliente chiamerà l'FM per fornire una risposta al cliente nella fase 3a. La risposta generata dalla FM verrà quindi archiviata come vettore in MemoryDB per le query future (fase 3b) per ridurre al minimo i costi FM su domande semanticamente simili. In questo flusso, il passaggio 4 non verrebbe richiamato in quanto non esisteva una domanda semanticamente simile per la query originale.

Rilevamento di attività fraudolente

Il rilevamento delle frodi, una forma di rilevamento delle anomalie, rappresenta le transazioni valide come vettori confrontando le rappresentazioni vettoriali delle nuove transazioni nette. La frode viene rilevata quando queste nuove transazioni nette hanno una bassa somiglianza con i vettori che rappresentano i dati transazionali validi. Ciò consente di rilevare le frodi modellando il comportamento normale, anziché cercare di prevedere ogni possibile caso di frode. MemoryDB consente alle organizzazioni di eseguire questa operazione in periodi di elevata produttività, con falsi positivi minimi e una latenza di un millisecondo.



Altri casi d'uso

- I motori di raccomandazione possono trovare agli utenti prodotti o contenuti simili rappresentando gli elementi come vettori. I vettori vengono creati analizzando attributi e modelli. In base ai modelli e agli attributi degli utenti, è possibile consigliare agli utenti nuovi elementi invisibili trovando i vettori più simili già valutati positivamente all'utente.

- I motori di ricerca di documenti rappresentano i documenti di testo come vettori densi di numeri, che catturano il significato semantico. Al momento della ricerca, il motore converte una query di ricerca in un vettore e trova i documenti con i vettori più simili alla query utilizzando la ricerca approssimativa del vicino più prossimo. Questo approccio alla somiglianza vettoriale consente di abbinare i documenti in base al significato anziché semplicemente alle parole chiave.

Utilizzando il AWS Management Console

Questa funzionalità è disponibile in anteprima per MemoryDB for Redis ed è soggetta a modifiche.

Per creare un cluster abilitato alla ricerca vettoriale all'interno della console, è necessario abilitare la ricerca vettoriale nelle impostazioni del cluster. La ricerca vettoriale è disponibile per MemoryDB for Redis versione 7.1 e una configurazione a singolo shard.

Cluster settings

- Enable vector search - *public preview* [Info](#)
You can store vector embeddings and perform vector searches.

i The preview for vector search is compatible with MemoryDB for Redis version 7.1 and a single shard configuration. Vector search and these configurations cannot be modified after creation. We recommend you do not enable this for production clusters.

Per ulteriori informazioni sull'utilizzo della ricerca vettoriale con, vedere. [AWS Management Console Creazione di un cluster \(Console\)](#)

Utilizzando il AWS Command Line Interface

Questa funzionalità è disponibile in anteprima per MemoryDB for Redis ed è soggetta a modifiche.

Per creare un cluster MemoryDB abilitato alla ricerca vettoriale, è possibile utilizzare il comando MemoryDB [create-cluster](#) passando un gruppo di parametri immutabile per abilitare la modalità di anteprima per le funzionalità di ricerca vettoriale. `default.memorydb-redis7.search.preview`

```
aws memorydb create-cluster \  
  --cluster-name <value> \  
  --node-type <value> \  
  --search-enabled <value>
```

```
--engine redis \  
--engine-version 7.1 \  
--num-shards 1 \  
--acl-name <value> \  
--parameter-group-name default.memorydb-redis7.search.preview
```

Comandi di ricerca vettoriale

Di seguito è riportato un elenco di comandi supportati per la ricerca vettoriale.

Argomenti

- [FT.CREATE](#)
- [FT.SEARCH](#)
- [FT.AGGREGATE](#)
- [FT.DROPINDEX](#)
- [FT.INFO](#)
- [FT._LISTA](#)
- [FT.ALIASADD](#)
- [FT.ALIASDEL](#)
- [FT.ALIASUPDATE](#)
- [FT._LISTA DI ALIAS](#)
- [FT.CONFIG OTTIENI](#)
- [AIUTO FT.CONFIG](#)
- [SET FT.CONFIG](#)
- [FT.PROFILE](#)
- [FT.EXPLAIN](#)
- [FT.EXPLAINCLI](#)

FT.CREATE

Crea un indice e avvia un riempimento di tale indice. Per ulteriori informazioni, consulta [Panoramica della ricerca vettoriale](#) per i dettagli sulla costruzione dell'indice.

Sintassi

```

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>])
)+

```

Schema

- **Identificatore di campo:**
 - Per le chiavi hash, l'identificatore di campo è un nome di campo.
 - Per le chiavi JSON, l'identificatore di campo è un percorso JSON.

Per ulteriori informazioni, consulta [Tipi di campi dell'indice](#).

- **Tipi di campo:**
 - **TAG:** per ulteriori informazioni, consulta [Tag](#).
 - **NUMERICO:** Il campo contiene un numero.
 - **TESTO:** Il campo contiene qualsiasi blob di dati.
 - **VECTOR:** campo vettoriale che supporta la ricerca vettoriale.
 - **Algoritmo:** può essere HNSW (Hierarchical Navigable Small World) o FLAT (forza bruta).
 - **attr_count**— numero di attributi che verranno passati come configurazione dell'algoritmo, inclusi sia nomi che valori.
 - **{attribute_name} {attribute_value}**— coppie chiave/valore specifiche dell'algoritmo che definiscono la configurazione dell'indice.

Per l'algoritmo FLAT, gli attributi sono:

Campo obbligatorio:

- **DIM** — Numero di dimensioni nel vettore.

- `DISTANCE_METRIC` — Può essere uno dei [L2 | IP | COSINE].
- `TYPE` — Tipo di vettore. L'unico tipo supportato è `FLOAT32`.

Facoltativo:

- `INITIAL_CAP` — La capacità vettoriale iniziale dell'indice influisce sulla dimensione di allocazione della memoria dell'indice.

Per l'algoritmo HNSW, gli attributi sono:

Campo obbligatorio:

- `TIPO`: tipo vettoriale. L'unico tipo supportato è `FLOAT32`.
- `DIM`: dimensione vettoriale, specificata come numero intero positivo. Massimo: 32768
- `DISTANCE_METRIC` — Può essere uno dei [L2 | IP | COSINE].

Facoltativo:

- `INITIAL_CAP` — La capacità vettoriale iniziale dell'indice influisce sulla dimensione di allocazione della memoria dell'indice. Il valore predefinito è 1024.
- `M` — Numero massimo di bordi in uscita consentiti per ogni nodo del grafico in ogni livello. Sul livello zero il numero massimo di bordi in uscita sarà $2M$. Il valore predefinito è 16. Il massimo è 512.
- `EF_CONSTRUCTION` — controlla il numero di vettori esaminati durante la costruzione dell'indice. Valori più elevati per questo parametro miglioreranno il rapporto di richiamo a scapito di tempi più lunghi di creazione dell'indice. Il valore predefinito è 200. Il valore massimo è 4096.
- `EF_RUNTIME`: controlla il numero di vettori esaminati durante le operazioni di interrogazione. Valori più elevati per questo parametro possono migliorare il richiamo a scapito di tempi di interrogazione più lunghi. Il valore di questo parametro può essere sovrascritto in base alla singola query. Il valore predefinito è 10. Il valore massimo è 4096.

Valori restituiti

Restituisce una semplice stringa di messaggio OK o una risposta all'errore.

Examples (Esempi)

Note

L'esempio seguente utilizza argomenti nativi di [redis-cli](#), come la dequotazione e l'eliminazione dell'escape dei dati, prima di inviarli a Redis. Per utilizzare altri client in linguaggi di programmazione (Python, Ruby, C#, ecc.), segui le regole di gestione di tali ambienti per la gestione di stringhe e dati binari. [Per ulteriori informazioni sui client supportati, consulta Strumenti su cui costruire AWS](#)

Example 1: Crea alcuni indici**Crea un indice per vettori di dimensione 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
```

Crea un indice JSON a 6 dimensioni utilizzando l'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 Esempio 2: popolare alcuni dati

I seguenti comandi sono formattati in modo da poter essere eseguiti come argomenti del programma terminale redis-cli. Gli sviluppatori che utilizzano client in linguaggio di programmazione (come Python, Ruby, C#, ecc.) dovranno seguire le regole di gestione del loro ambiente per gestire stringhe e dati binari.

Creazione di alcuni dati hash e json:

```
HSET hash:0 vec "\x00\x00\x00\x00\x00\x00\x00\x00"
HSET hash:1 vec "\x00\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]}'
```

Tieni presente quanto segue:

FT.SEARCH

Utilizza l'espressione di interrogazione fornita per individuare le chiavi all'interno di un indice. Una volta individuati, è possibile restituire il conteggio e/o il contenuto dei campi indicizzati all'interno di tali chiavi. Per ulteriori informazioni, vedere [Espressione di interrogazione di ricerca vettoriale](#).

Per creare dati da utilizzare in questi esempi, vedete il comando [FT.CREATE](#).

Sintassi

```
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:** Questa clausola identifica quali campi di una chiave vengono restituiti. La clausola **AS** opzionale su ogni campo sostituisce il nome del campo nel risultato. È possibile specificare solo i campi che sono stati dichiarati per questo indice.
- **LIMIT: <offset><count>:** Questa clausola fornisce la funzionalità di impaginazione in quanto vengono restituite solo le chiavi che soddisfano i valori di offset e count. Se questa clausola viene omessa, il valore predefinito è «LIMIT 0 10», ovvero verranno restituite solo un massimo di 10 chiavi.
- **PARAMETRI:** due volte il numero di coppie chiave-valore. È possibile fare riferimento alle coppie chiave/valore del parametro dall'interno dell'espressione di query. Per ulteriori informazioni, vedete [Espressione di interrogazione di ricerca vettoriale](#).
- **COUNT:** questa clausola elimina la restituzione del contenuto delle chiavi, viene restituito solo il numero di chiavi. Questo è un alias per «LIMIT 0 0».

Valori restituiti

Restituisce una matrice o una risposta di errore.

- Se l'operazione viene completata correttamente, restituisce un array. Il primo elemento è il numero totale di chiavi corrispondenti alla query. Gli elementi rimanenti sono coppie di nomi di chiavi ed elenchi di campi. L'elenco dei campi è un altro array che comprende coppie di nomi di campo e valori.

- Se l'indice è in corso di riempimento, il comando restituisce immediatamente una risposta di errore.
- Se viene raggiunto il timeout, il comando restituisce una risposta di errore.

Esempio: esegui alcune ricerche

Note

L'esempio seguente utilizza argomenti nativi di [redis-cli](#), come la dequotazione e la cancellazione dei dati, prima di inviarli a Redis. Per utilizzare altri client in linguaggi di programmazione (Python, Ruby, C#, ecc.), segui le regole di gestione di tali ambienti per la gestione di stringhe e dati binari. [Per ulteriori informazioni sui client supportati, consulta Strumenti su cui costruire AWS](#)

Una ricerca 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"
```

Ciò produce due risultati, ordinati in base al punteggio, che è la distanza dal vettore di query (immessa come esadecimale).

Ricerche 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\x00"
DIALECT 2
1) (integer) 2
2) "json:2"
```



```
[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]
```

- Le clausole FILTER, LIMIT, GROUPBY, SORTBY e APPLY possono essere ripetute più volte in qualsiasi ordine e possono essere mescolate liberamente. Vengono applicate nell'ordine specificato con l'output di una clausola che alimenta l'input della clausola successiva.
- Nella sintassi precedente, una «proprietà» è un campo dichiarato nel comando [FT.CREATE](#) per questo indice OPPURE l'output di una precedente clausola APPLY o funzione REDUCE.
- La clausola LOAD si limita al caricamento di campi dichiarati nell'indice. «LOAD *» caricherà tutti i campi dichiarati nell'indice.
- Sono supportate le seguenti funzioni di riduzione: COUNT, COUNT_DISTINCTISH, SUM, MIN, MAX, AVG, STDDEV, QUANTILE, TOLIST, FIRST_VALUE e RANDOM_SAMPLE. [Per ulteriori informazioni, vedere Aggregazioni](#)
- LIMITE <offset><count>: Conserva i record a partire da <offset> e fino a <count>, tutti gli altri record vengono eliminati.
- PARAMETRI: due volte il numero di coppie chiave-valore. È possibile fare riferimento alle coppie chiave/valore del parametro dall'interno dell'espressione di query. Per ulteriori informazioni, vedete [Espressione di interrogazione di ricerca vettoriale](#).

Valori restituiti

Restituisce una matrice o una risposta di errore.

- Se l'operazione viene completata correttamente, restituisce un array. Il primo elemento è un numero intero senza un significato particolare (deve essere ignorato). Gli elementi rimanenti sono i risultati prodotti dall'ultima fase. Ogni elemento è una matrice di coppie di nomi di campo e valori.
- Se l'indice è in corso di riempimento, il comando restituisce immediatamente una risposta di errore.
- Se viene raggiunto il timeout, il comando restituisce una risposta di errore.

FT.DROPINDEX

Eliminare un indice. La definizione dell'indice e il contenuto associato vengono eliminati. Le chiavi Redis non sono interessate.

Sintassi

```
FT.DROPINDEX <index-name>
```

Valori restituiti

Restituisce una semplice stringa, un messaggio OK o una risposta di errore.

FT.INFO

Sintassi

```
FT.INFO <index-name>
```

L'output della pagina FT.INFO è un array di coppie chiave-valore, come descritto nella tabella seguente:

Chiave	Value type (Tipo di valore)	Descrizione
nome_indice	string	Nome dell'indice
creation_timestamp	integer	Timestamp dell'ora di creazione in stile Unix
tipo_chiave	string	HASH o JSON
key_prefixes	matrice di stringhe	Prefissi chiave per questo indice
campi	matrice di informazioni sul campo	Campi di questo indice
space_usage	integer	Byte di memoria utilizzati da questo indice

Chiave	Value type (Tipo di valore)	Descrizione
fullext_space_usage	integer	Byte di memoria utilizzati dai campi non vettoriali
vector_space_usage	integer	Byte di memoria usati dai campi vettoriali
num_docs	integer	Numero di chiavi attualmente contenute nell'indice
num_indexed_vectors	integer	Numero di vettori attualmente contenuti nell'indice
current_lag	integer	Ritardo di ingestione recente (millisecondi)
backfill_status	string	Uno dei seguenti: Completato, Sospeso o InProgress Non riuscito

La tabella seguente descrive le informazioni per ogni campo:

Chiave	Value type (Tipo di valore)	Descrizione
identificatore	string	nome del campo
field_name	string	Nome del membro hash o percorso JSON
type (tipo)	string	uno tra: Numerico, Tag, Testo o Vettore
option	string	ignora

Se il campo è di tipo Vector, saranno presenti informazioni aggiuntive a seconda dell'algoritmo.

Per l'algoritmo HNSW:

Chiave	Value type (Tipo di valore)	Descrizione
Algoritmo	string	HNSW
data_type	string	GALLEGGIANTE 32
distanza_metrica	string	uno tra: L2, IP o Cosine
capacità_iniziale	integer	Dimensione iniziale dell'indice del campo vettoriale
capacità_corrente	integer	Dimensione attuale dell'indice dei campi vettoriali
massimum_edges	integer	Parametro M alla creazione
ef_construction	integer	Parametro EF_CONSTRUCTION alla creazione
ef_runtime	integer	Parametro EF_RUNTIME alla creazione

Per l'algoritmo FLAT:

Chiave	Value type (Tipo di valore)	Descrizione
Algoritmo	string	APPARTAMENTO
data_type	string	GALLEGGIANTE 32
distanza_metrica	string	uno tra: L2, IP o Cosine
capacità_iniziale	integer	Dimensione iniziale dell'indice del campo vettoriale
capacità_corrente	integer	Dimensione attuale dell'indice dei campi vettoriali

FT._LISTA

Elenca tutti gli indici.

Sintassi

```
FT._LIST
```

Valori restituiti

Restituisce una matrice di nomi di indici

FT.ALIASADD

Aggiungi un alias per un indice. Il nuovo nome alias può essere utilizzato ovunque sia richiesto un nome di indice.

Sintassi

```
FT.ALIASADD <alias> <index-name>
```

Valori restituiti

Restituisce un semplice messaggio OK in formato stringa o una risposta di errore.

FT.ALIASDEL

Elimina un alias esistente per un indice.

Sintassi

```
FT.ALIASDEL <alias>
```

Valori restituiti

Restituisce un semplice messaggio OK in formato stringa o una risposta di errore.

FT.ALIASUPDATE

Aggiorna un alias esistente in modo che punti a un indice fisico diverso. Questo comando ha effetto solo sui riferimenti futuri all'alias. Le operazioni attualmente in corso (FT.SEARCH, FT.AGGREGATE) non sono influenzate da questo comando.

Sintassi

```
FT.ALIASUPDATE <alias> <index>
```

Valori restituiti

Restituisce una semplice stringa di messaggio OK o una risposta di errore.

FT. _LISTA DI ALIAS

Elenca gli alias dell'indice.

Sintassi

```
FT._ALIASLIST
```

Valori restituiti

Restituisce un array della dimensione del numero di alias correnti. Ogni elemento dell'array è la coppia alias-indice.

FT.CONFIG OTTIENI

Restituisce il valore del parametro TIMEOUT.

Sintassi

```
FT.CONFIG GET [* | <timeout>]
```

Valori restituiti

Restituisce il valore del parametro TIMEOUT.

AIUTO FT.CONFIG

Recupera informazioni sul parametro TIMEOUT.

Sintassi

```
FT.CONFIG HELP [* | <timeout>]
```

Valori restituiti

Restituisce informazioni sul parametro TIMEOUT

SET FT.CONFIG

Imposta il parametro TIMEOUT. Il valore predefinito è 10.000 millisecondi.

Note

I nomi dei parametri configurabili non fanno distinzione tra maiuscole e minuscole.

Sintassi

```
FT.CONFIG SET <timeout> <value>
```

Valori restituiti

Restituisce il valore dell'impostazione TIMEOUT.

FT.PROFILE

Esegui una query e restituisci le informazioni sul profilo relative a tale query.

Sintassi

```
FT.PROFILE  
  
<index>  
SEARCH | AGGREGATE  
[LIMITED]  
QUERY <query ....>
```

Valori restituiti

Un array a due elementi. Il primo elemento è il risultato del FT . AGGREGATE comando FT . SEARCH or a cui è stato profilato. Il secondo elemento è una serie di informazioni sulle prestazioni e sulla profilazione.

FT.EXPLAIN

Analizza una query e restituisce informazioni su come tale query è stata analizzata.

Sintassi

```
FT.EXPLAIN <index> <query>
```

Valori restituiti

Una stringa contenente i risultati analizzati.

FT.EXPLAINCLI

Uguale al comando FT.EXPLAIN tranne per il fatto che i risultati vengono visualizzati in un formato diverso, più utile con redis-cli.

Sintassi

```
FT.EXPLAINCLI <index> <query>
```

Valori restituiti

Una stringa contenente i risultati analizzati.

Sicurezza in MemoryDB for Redis

Per AWS, la sicurezza del cloud ha la massima priorità. In quanto cliente AWS, è possibile trarre vantaggio da un'architettura di data center e di rete progettata per soddisfare i requisiti delle organizzazioni più esigenti a livello di sicurezza.

La sicurezza è una responsabilità condivisa tra te e AWS. Il [modello di responsabilità condivisa](#) descrive questo modello come sicurezza del cloud e sicurezza nel cloud:

- La sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che esegue AWS i servizi nel AWS Cloud. AWS fornisce, inoltre, servizi utilizzabili in modo sicuro. I revisori di terze parti testano regolarmente e verificano l'efficacia della nostra sicurezza nell'ambito dei [Programmi di conformità AWS](#). Per informazioni sui programmi di conformità applicabili a MemoryDB, consulta [AWS Services in Scope by Compliance Program \(Servizi coperti dal programma di conformità\) Services in Scope by Compliance Program](#) ().
- Sicurezza nel cloud: la tua responsabilità è determinata dal servizio AWS che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i tuoi requisiti aziendali e le leggi e le normative applicabili

Questa documentazione consente di comprendere come applicare il modello di responsabilità condivisa quando si usa MemoryDB for Redis. Viene illustrato come configurare MemoryDB per soddisfare gli obiettivi di sicurezza e conformità. È inoltre illustrato come utilizzare altri AWS servizi che consentono di monitorare e proteggere le risorse MemoryDB.

Indice

- [Protezione dei dati in MemoryDB per Redis](#)
- [Gestione delle identità e degli accessi in MemoryDB per Redis](#)
- [Registrazione di log e monitoraggio](#)
- [Convalida della conformità per MemoryDB for Redis](#)
- [Sicurezza dell'infrastruttura in Amazon MemoryDL](#) [la sicurezza dell'infrastruttura in Amazon MemoryDL](#)
- [Riservatezza del traffico Internet](#)
- [Aggiornamenti del servizio in MemoryDB per Redis](#)

Protezione dei dati in MemoryDB per Redis

Il modello di [responsabilità AWS condivisa modello](#) di di si applica alla protezione dei dati in. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che esegue tutto l'Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. Inoltre, sei responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS che utilizzi. Per ulteriori informazioni sulla privacy dei dati, vedi le [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al [Modello di responsabilità condivisa AWS e GDPR](#) nel Blog sulla sicurezza AWS.

Per garantire la protezione dei dati, ti suggeriamo di proteggere le credenziali Account AWS e di configurare singoli utenti con AWS IAM Identity Center o AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Utilizza SSL/TLS per comunicare con le risorse AWS. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail.
- Utilizza le soluzioni di crittografia AWS, insieme a tutti i controlli di sicurezza predefiniti in Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se necessiti di moduli crittografici convalidati FIPS 140-2 quando accedi ad AWS attraverso un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-2](#).

Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori o Servizi AWS utilizzi la console, l'API o AWS gli SDK. AWS CLI I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

Sicurezza dei dati in MemoryDB per Redis

Per aiutarti a proteggere i tuoi dati, MemoryDB for Redis e Amazon EC2 forniscono meccanismi di protezione contro l'accesso non autorizzato ai tuoi dati sul server.

MemoryDB fornisce anche funzionalità di crittografia per i dati sui cluster:

- La crittografia dei dati in transito esegue la crittografia dei dati quando si spostano da una posizione a un'altra, ad esempio tra i nodi nel cluster o tra il cluster e l'applicazione.
- La crittografia At-Rest crittografa il registro delle transazioni e i dati su disco durante le operazioni di snapshot.

Puoi anche utilizzarla [Autenticazione degli utenti con elenchi di controllo degli accessi \(ACL\)](#) per controllare l'accesso degli utenti ai tuoi cluster.

Argomenti

- [Crittografia dei dati inattivi in MemoryDB](#)
- [Crittografia in transito \(TLS\) in MemoryDB](#)
- [Autenticazione degli utenti con elenchi di controllo degli accessi \(ACL\)](#)
- [Autenticazione con IAM](#)

Crittografia dei dati inattivi in MemoryDB

Per mantenere la sicurezza dei dati, MemoryDB for Redis e Amazon S3 forniscono metodi diversi per limitare l'accesso ai dati nei cluster. Per ulteriori informazioni, consultare [MemoryDB e Amazon VPC](#) e [Gestione delle identità e degli accessi in MemoryDB per Redis](#).

La crittografia dei dati inattivi di MemoryDB è sempre abilitata per incrementare la sicurezza dei dati crittografando i dati persistenti. Crittografa i seguenti aspetti:

- Dati nel registro delle transazioni
- Disco durante le operazioni di sincronizzazione, istantanea e swap
- Istantanee archiviate in Amazon S3

MemoryDB offre la crittografia di default (servizio gestito) dei dati inattivi, nonché la possibilità di utilizzare le chiavi simmetriche root gestite del cliente in [AWSKey Management Service \(KMS\)](#).

I dati archiviati su SSD (Solid State Drive) in cluster abilitati per il tiering di dati sono sempre crittografati per impostazione predefinita.

Per informazioni sulla crittografia dei dati in transito, consulta [Crittografia in transito \(TLS\) in MemoryDB](#)

Argomenti

- [Utilizzo delle chiavi gestite dai clienti diAWS KMS](#)
- [Vedi anche](#)

Utilizzo delle chiavi gestite dai clienti diAWS KMS

MemoryDB supporta le chiavi simmetriche root gestite dal cliente (chiave KMS) per la crittografia inattiva. Le chiavi KMS gestite dal cliente sono chiavi crittografiche che è possibile creare, possedere e gestire all'interno del proprio account AWS. Per ulteriori informazioni, consulta [Customer RootAWS Key](#) nella Guida per gli sviluppatori di Key Management Service. Le chiavi devono essere create inAWS KMS prima di poter essere utilizzate con MemoryDB.

Per informazioni su come creare chiavi master KMS AWS, consulta [Creazione di chiavi](#) nella AWSGuida per gli sviluppatori di Key Management Service.

MemoryDB consente l'integrazione con AWS KMS. Per ulteriori informazioni, consulta [Utilizzo di concessioni](#) nella AWS Guida per gli sviluppatori Key Management Service. Non è necessaria nessuna operazione per abilitare l'integrazione MemoryDB con AWS KMS.

La chiave `kms:ViaService` condiziona l'uso di una chiave AWS KMS alle richieste da AWS servizi specificati. Da utilizzare `kms:ViaService` con MemoryDB, includi entrambi `ViaService` i nomi nel valore della chiave della condizione: `memorydb.amazonaws.com`. Per ulteriori informazioni, consulta [kms:ViaService](#).

Puoi utilizzarlo [AWS CloudTrail](#) per tenere traccia delle richieste che MemoryDB for Redis invia a per AWS Key Management Service conto tuo. Tutte le chiamate API a AWS Key Management Service riguardo le chiavi gestite dal cliente hanno CloudTrail registri corrispondenti. Puoi anche vedere le sovvenzioni create da MemoryDB chiamando la chiamata API [ListGrants](#) KMS.

Dopo che un cluster viene crittografato utilizzando una chiave gestita dal cliente, tutte le istantanee del cluster vengono crittografate nel modo seguente:

- Le istantanee automatiche giornaliere vengono crittografate utilizzando la chiave gestita dal cliente associata al cluster.
- Anche l'istantanea finale creata al momento dell'eliminazione del cluster viene crittografata utilizzando la chiave gestita dal cliente associata al cluster.
- Le istantanee create manualmente vengono crittografate per impostazione predefinita per utilizzare la chiave KMS associata al cluster. Puoi sostituirla scegliendo un'altra CMK gestita dal cliente.
- La copia di un'istantanea viene impostata in modo di default sull'uso della chiave gestita dal cliente associata all'istantanea fonte. Puoi sostituirla scegliendo un'altra CMK gestita dal cliente.

Note

- Le chiavi gestite dal cliente non possono essere utilizzate durante l'esportazione delle istantanee sul bucket Amazon S3 selezionato. Tuttavia, tutte le istantanee esportate in Amazon S3 vengono crittografate utilizzando la [crittografia lato server](#). Puoi scegliere di copiare i file istantanei su un nuovo oggetto S3 e crittografarli utilizzando una chiave gestita dal cliente, copiare i file in un altro bucket S3 configurato con la crittografia predefinita mediante una chiave KMS o modificare l'opzione di crittografia nel file stesso.
- Puoi anche utilizzare le chiavi gestite dal cliente per crittografare le istantanee create manualmente che non utilizzano chiavi gestite dal cliente per la crittografia. Con questa

opzione, il file di istantanea archiviato in Amazon S3 viene crittografato utilizzando una chiave KMS, anche se i dati non sono crittografati sul cluster di fonte.

Il ripristino da un'istantanea consente di scegliere tra le opzioni di crittografia disponibili, simili alle opzioni di crittografia disponibili durante la creazione di un nuovo cluster.

- Se si elimina la chiave o si [disattiva](#) la chiave e si [revocano le concessioni](#) per la chiave utilizzata per crittografare un cluster, il cluster diventa irrecuperabile. In altre parole, non può essere modificato o recuperato dopo un errore hardware. AWS KMS elimina le chiavi master solo dopo un periodo di attesa di almeno sette giorni. Dopo che la chiave è stata eliminata, puoi utilizzare una chiave gestita dal cliente differente per creare un'istantanea per scopi di archiviazione.
- La rotazione automatica delle chiavi permette di mantenere le proprietà delle chiavi rootAWS KMS e di conseguenza non ha effetto sulla tua capacità di accedere ai dati MemoryDB. I cluster MemoryDB crittografati non supportano la rotazione manuale delle chiavi, che comporta la creazione di una nuova chiave root e l'aggiornamento di tutti i riferimenti alla chiave precedente. Per ulteriori informazioni, consulta [Rotazione delle chiavi root del cliente](#) nella GuidaAWS per gli sviluppatori di Key Management Service.
- La crittografia di un cluster MemoryDB utilizzando la chiave KMS richiede una concessione per cluster. Questa sovvenzione viene utilizzata per tutta la durata del cluster. Inoltre, durante la creazione dell'istantanea viene utilizzata una concessione per istantanea. Questa concessione viene ritirata una volta creata l'istantanea.
- Per ulteriori informazioni su concessioni e limitiAWS KMS, consulta [Quote](#) nella Guida per gli sviluppatori diAWS Key Management Service.

Vedi anche

- [Crittografia in transito \(TLS\) in MemoryDB](#)
- [MemoryDB e Amazon VPC](#)
- [Gestione delle identità e degli accessi in MemoryDB per Redis](#)

Crittografia in transito (TLS) in MemoryDB

Per aiutarti a proteggere i tuoi dati, MemoryDB for Redis e Amazon EC2 forniscono meccanismi di protezione contro l'accesso non autorizzato ai tuoi dati sul server. Fornendo funzionalità di crittografia

in transito, MemoryDB ti offre uno strumento che puoi utilizzare per proteggere i tuoi dati quando vengono spostati da una posizione all'altra. Ad esempio, è possibile spostare i dati da un nodo primario a un nodo di replica di lettura all'interno di un cluster o tra il cluster e l'applicazione.

Argomenti

- [Panoramica della crittografia dei dati in transito](#)
- [Consulta anche](#)

Panoramica della crittografia dei dati in transito

La crittografia in transito di MemoryDB for Redis è una funzionalità che aumenta la sicurezza dei dati nei punti più vulnerabili, quando sono in transito da una posizione all'altra.

La crittografia in transito di MemoryDB implementa le seguenti funzionalità:

- Connessioni crittografate: sia le connessioni server che quelle client sono crittografate con Transport Layer Security (TLS).
- Replica crittografata: i dati che si spostano tra un nodo primario e nodi di replica vengono crittografati.
- Autenticazione del server: i client possono autenticare che si stanno connettendo al server giusto.

A partire dal 20/07/2023, TLS 1.2 è la versione minima supportata per i cluster nuovi ed esistenti. Usa questo [link](#) per saperne di più su TLS 1.2 all'indirizzo. AWS

Per ulteriori informazioni sulla connessione ai cluster MemoryDB, vedere. [Connessione ai nodi MemoryDB utilizzando redis-cli](#)

Consulta anche

- [Crittografia dei dati inattivi in MemoryDB](#)
- [Autenticazione degli utenti con elenchi di controllo degli accessi \(ACL\)](#)
- [MemoryDB e Amazon VPC](#)
- [Gestione delle identità e degli accessi in MemoryDB per Redis](#)

Autenticazione degli utenti con elenchi di controllo degli accessi (ACL)

È possibile autenticare gli utenti con gli elenchi di controllo degli accessi (ACL).

Gli ACL consentono di controllare l'accesso al cluster raggruppando gli utenti. Queste liste di controllo degli accessi sono progettate per organizzare l'accesso ai cluster.

Con gli ACL, è possibile creare utenti e assegnare loro autorizzazioni specifiche utilizzando una stringa di accesso, come descritto nella sezione successiva. Gli utenti vengono assegnati agli elenchi di controllo degli accessi allineati a un ruolo specifico (amministratori, risorse umane) che vengono quindi distribuiti in uno o più cluster di MemoryDB. In questo modo, è possibile stabilire limiti di sicurezza tra i client che utilizzano lo stesso cluster o gli stessi cluster di MemoryDB e impedire ai client di accedere ai dati degli altri.

Gli ACL sono progettati per supportare l'introduzione di [Redis ACL in Redis 6](#). Quando si utilizzano gli ACL con il cluster MemoryDB, esistono alcune limitazioni:

- Non è possibile specificare password in una stringa di accesso. Le password vengono impostate con o chiamate. [CreateUserUpdateUser](#)
- Per i diritti utente, si passa `on` o `off` come parte della stringa di accesso. Se nessuno dei due è specificato nella stringa di accesso, l'utente viene assegnato `off` e non dispone dei diritti di accesso al cluster.
- Non puoi usare comandi proibiti. Se specifichi un comando proibito, verrà generata un'eccezione. Per un elenco di questi comandi, vedi [Comandi Redis limitati](#).
- Non è possibile utilizzare `reset` come parte di una stringa di accesso. Si specificano le password con parametri API e MemoryDB gestisce le password. Pertanto, non è possibile utilizzare `reset` perché rimuoverebbe tutte le password per un utente.
- Redis 6 introduce il comando [ELENCO ACL](#). Questo comando restituisce un elenco di utenti insieme alle regole ACL applicate a ciascun utente. MemoryDB supporta il `ACL LIST` comando, ma non include il supporto per gli hash delle password come fa Redis. Con MemoryDB, è possibile utilizzare l'[DescribeUsers](#) operazione per ottenere informazioni simili, incluse le regole contenute nella stringa di accesso. Tuttavia, [DescribeUsers](#) non recupera una password utente.

[Altri comandi di sola lettura supportati da MemoryDB includono ACL WHOAMI, ACL USERS e ACL CAT](#). MemoryDB non supporta nessun altro comando ACL basato sulla scrittura.

L'uso degli ACL con MemoryDB è descritto più dettagliatamente di seguito.

Argomenti

- [Specifica delle autorizzazioni mediante una stringa di accesso](#)
- [Funzionalità di ricerca vettoriale](#)

- [Applicazione degli ACL a un cluster per MemoryDB](#)

Specifica delle autorizzazioni mediante una stringa di accesso

Per specificare le autorizzazioni per un cluster MemoryDB, si crea una stringa di accesso e la si assegna a un utente, utilizzando o. AWS CLI o AWS Management Console

Le stringhe di accesso sono definite come un elenco di regole delimitate da spazi che vengono applicate all'utente. Essi definiscono quali comandi un utente può eseguire e quali chiavi un utente può operare. Per eseguire un comando, un utente deve avere accesso al comando in esecuzione e tutte le chiavi sono accessibili dal comando. Le regole vengono applicate cumulativamente da sinistra a destra e, se nella stringa fornita sono presenti ridondanze, è possibile utilizzare una stringa più semplice al posto di quella fornita.

Per ulteriori informazioni sulla sintassi delle regole ACL, consulta [ACL](#).

Nell'esempio seguente, la stringa di accesso rappresenta un utente attivo con accesso a tutti i tasti e i comandi disponibili.

```
on ~* &* +@all
```

La sintassi della stringa di accesso è suddivisa come segue:

- `on`— L'utente è un utente attivo.
- `~*`— L'accesso è dato a tutte le chiavi disponibili.
- `&*`— L'accesso è dato a tutti i canali pubsub.
- `+@all`— Accesso a tutti i comandi disponibili.

Le impostazioni precedenti sono le meno restrittive. È possibile modificare queste impostazioni per renderle più sicure.

Nell'esempio seguente, la stringa di accesso rappresenta un utente con accesso limitato all'accesso in lettura sulle chiavi che iniziano con lo spazio delle chiavi «app::»

```
on ~app::* -@all +@read
```

È possibile perfezionare ulteriormente queste autorizzazioni elencando i comandi a cui l'utente ha accesso:

`+command1`— L'accesso dell'utente ai comandi è limitato a `command1`.

+@category— L'accesso dell'utente è limitato a una categoria di comandi.

Per informazioni sull'assegnazione di una stringa di accesso a un utente, vedere [Creazione di utenti ed elenchi di controllo degli accessi con la console e la CLI](#).

Se stai migrando un carico di lavoro esistente su MemoryDB, puoi recuperare la stringa di accesso chiamando `ACL LIST`, escludendo l'utente e qualsiasi hash della password.

Funzionalità di ricerca vettoriale

Note

Questa funzionalità è disponibile in anteprima per MemoryDB for Redis ed è soggetta a modifiche.

Infatti [Ricerca vettoriale](#), tutti i comandi di ricerca appartengono alla @search categoria e alle categorie @read esistenti @fast e @slow vengono aggiornati per includere i comandi di ricerca. @write Se un utente non ha accesso a una categoria, non ha accesso a nessun comando all'interno della categoria. Ad esempio, se l'utente non ha accesso a @search, non può eseguire alcun comando relativo alla ricerca.

La tabella seguente indica la mappatura dei comandi di ricerca alle categorie appropriate.

Comandi 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

Comandi VSS	@read	@write	@fast	@slow
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	
FT.CONFIG	Y		Y	

Applicazione degli ACL a un cluster per MemoryDB

Per utilizzare gli ACL di MemoryDB, procedi nel seguente modo:

1. Crea uno o più utenti.
2. Crea un ACL e aggiungi utenti all'elenco.
3. Assegna l'ACL a un cluster.

La tabella seguente descrive i seguenti passaggi nel dettaglio.

Argomenti

- [Creazione di utenti ed elenchi di controllo degli accessi con la console e la CLI](#)
- [Gestione degli elenchi di controllo degli accessi con la console e la CLI](#)
- [Assegnazione delle liste di controllo degli accessi ai cluster](#)

Creazione di utenti ed elenchi di controllo degli accessi con la console e la CLI

Le informazioni utente per gli utenti ACL sono un nome utente e, facoltativamente, una password e una stringa di accesso. La stringa di accesso fornisce il livello di autorizzazione per i tasti e i comandi. Il nome è univoco per l'utente ed è quello che viene passato al motore.

Assicurati che le autorizzazioni utente fornite corrispondano allo scopo previsto dell'ACL. Ad esempio, se crei un ACL chiamato `Administrators`, qualsiasi utente aggiunto a quel gruppo dovrebbe avere la stringa di accesso impostata per l'accesso completo a tasti e comandi. Per gli utenti di un e-commerce ACL, è possibile impostare le stringhe di accesso in sola lettura.

MemoryDB configura automaticamente un utente predefinito per account con un nome utente `"default"`. Non sarà associato a nessun cluster a meno che non venga aggiunto esplicitamente a un ACL. Non è possibile eliminare o modificare questo utente. Questo utente è progettato per la compatibilità con il comportamento di default delle versioni precedenti di Redis e dispone di una stringa di accesso che consente di chiamare tutti i comandi e accedere a tutte le chiavi.

Verrà creato un ACL immutabile «ad accesso aperto» per ogni account che contiene l'utente predefinito. Questa è l'unica ACL di cui l'utente predefinito può essere membro. Quando si crea un cluster, è necessario selezionare un ACL da associare al cluster. Sebbene sia possibile applicare l'ACL «ad accesso aperto» all'utente predefinito, consigliamo vivamente di creare un ACL con utenti con autorizzazioni limitate alle loro esigenze aziendali.

I cluster che non hanno TLS abilitato devono utilizzare l'ACL «ad accesso aperto» per fornire un'autenticazione aperta.

Gli ACL possono essere creati senza utenti. Un ACL vuoto non avrebbe accesso a un cluster e può essere associato solo a cluster abilitati per TLS.

Quando si crea un utente, è possibile impostare fino a due password. Quando si modifica una password, vengono mantenute tutte le connessioni esistenti ai cluster.

In particolare, tenete presente questi vincoli relativi alla password utente quando utilizzate gli ACL per MemoryDB:

- Le password devono essere da 16 a 128 caratteri stampabili.
- I seguenti caratteri non alfanumerici non sono consentiti: , " " / @.

Gestione degli utenti con la console e la CLI

Creazione di un utente (Console)

Per creare utenti sulla console

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Nel riquadro di navigazione a sinistra, scegli Utenti.
3. Scegli Crea utente
4. Nella pagina Crea utente, inserisci un nome.

I vincoli di denominazione dei cluster sono i seguenti:

- Devono contenere da 1 a 40 caratteri alfanumerici o trattini.
 - Devono iniziare con una lettera.
 - Non possono contenere due trattini consecutivi.
 - Non possono terminare con un trattino.
5. In Password, puoi inserire fino a due password.
 6. In Stringa di accesso, inserisci una stringa di accesso. La stringa di accesso imposta il livello di autorizzazione per le chiavi e i comandi consentiti all'utente.
 7. Per i tag, puoi facoltativamente applicare tag per cercare e filtrare gli utenti o tenere traccia AWS dei costi.
 8. Scegli Crea.

Creare un utente utilizzando il AWS CLI

Per creare un utente utilizzando la CLI

- Utilizzate il comando [create-user](#) per creare un utente.

Per Linux, macOS o Unix:

```
aws memorydb create-user \  
  --user-name user-name-1 \  
  --access-string "~objects:* ~items:* ~public:*" \  
  --authentication-mode \  
    Passwords="abc",Type=password
```

Per Windows:

```
aws memorydb create-user ^  
  --user-name user-name-1 ^  
  --access-string "~objects:* ~items:* ~public:*" ^  
  --authentication-mode \  
    Passwords="abc",Type=password
```

Modifica di un utente (Console)

Per modificare gli utenti sulla console

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Nel riquadro di navigazione a sinistra, scegli Utenti.
3. Scegli il pulsante di opzione accanto all'utente che desideri modificare, quindi scegli Azioni -> Modifica
4. Se desideri modificare una password, scegli il pulsante di opzione Modifica password. Nota che se hai due password, devi inserirle entrambe quando ne modifichi una.
5. Se stai aggiornando la stringa di accesso, inserisci quella nuova.
6. Scegli Modifica.

Modificare un utente utilizzando AWS CLI

Per modificare un utente utilizzando la CLI;

1. Usa il comando [update-user](#) per modificare un utente.

- Quando un utente viene modificato, gli elenchi di controllo degli accessi associati all'utente vengono aggiornati, insieme a tutti i cluster associati all'ACL. Tutte le connessioni esistenti vengono mantenute. Di seguito vengono mostrati gli esempi.

Per Linux, macOS o Unix:

```
aws memorydb update-user \  
  --user-name user-name-1 \  
  --access-string "~objects:* ~items:* ~public:*
```

Per Windows:

```
aws memorydb update-user ^  
  --user-name user-name-1 ^  
  --access-string "~objects:* ~items:* ~public:*
```

Visualizzazione dei dettagli dell'utente (Console)

Per visualizzare i dettagli dell'utente sulla console

- [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
- Nel riquadro di navigazione a sinistra, scegli Utenti.
- Scegli l'utente in Nome utente o utilizza la casella di ricerca per trovare l'utente.
- In Impostazioni utente puoi controllare la stringa di accesso, il numero di password, lo stato e l'Amazon Resource Name (ARN) dell'utente.
- In Access control lists (ACL) puoi controllare l'ACL a cui appartiene l'utente.
- In Tag puoi rivedere tutti i tag associati all'utente.

Visualizzazione dei dettagli dell'utente utilizzando il AWS CLI

Utilizzare il comando [describe-users](#) per visualizzare i dettagli di un utente.

```
aws memorydb describe-users \  
  --user-name my-user-name
```

Eliminazione di un utente (Console)

Per eliminare utenti dalla console

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. Nel riquadro di navigazione a sinistra, scegli Utenti.
3. Scegli il pulsante di opzione accanto all'utente che desideri modificare, quindi scegli Azioni -> Elimina
4. Per confermare, inserisci delete nella casella di testo di conferma, quindi scegli Elimina.
5. Per annullare, scegliere Cancel (Annulla).

Eliminazione di un utente tramite AWS CLI

Per eliminare un utente utilizzando la CLI;

- Utilizzare il comando [delete-user](#) per eliminare un utente.

L'account viene eliminato e rimosso da tutti gli elenchi di controllo degli accessi a cui appartiene. Di seguito è riportato un esempio.

Per Linux, macOS o Unix:

```
aws memorydb delete-user \  
--user-name user-name-2
```

Per Windows:

```
aws memorydb delete-user ^  
--user-name user-name-2
```

Gestione degli elenchi di controllo degli accessi con la console e la CLI

È possibile creare elenchi di controllo degli accessi per organizzare e controllare l'accesso degli utenti a uno o più cluster, come illustrato di seguito.

Utilizzare la procedura seguente per gestire gli elenchi di controllo degli accessi utilizzando la console.

Creazione di una lista di controllo degli accessi (ACL) (console)

Per creare un elenco di controllo degli accessi utilizzando la console

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. Nel riquadro di navigazione a sinistra, scegli Access control lists (ACL).
3. Scegli Crea ACL.
4. Nella pagina Crea lista di controllo di accesso (ACL), inserisci un nome ACL.

I vincoli di denominazione dei cluster sono i seguenti:

- Devono contenere da 1 a 40 caratteri alfanumerici o trattini.
 - Devono iniziare con una lettera.
 - Non possono contenere due trattini consecutivi.
 - Non possono terminare con un trattino.
5. In Utenti selezionati, effettuate una delle seguenti operazioni:
 - a. Crea un nuovo utente selezionando Crea utente
 - b. Aggiungi utenti scegliendo Gestisci, quindi selezionando gli utenti dalla finestra di dialogo Gestisci utenti e quindi selezionando Scegli.
 6. Per quanto riguarda i tag, puoi facoltativamente applicare tag per cercare e filtrare gli ACL o tenere traccia AWS dei costi.
 7. Scegli Crea.

Creazione di una lista di controllo degli accessi (ACL) utilizzando il AWS CLI

Utilizzare le seguenti procedure per creare un elenco di controllo degli accessi utilizzando la CLI.

Per creare un nuovo ACL e aggiungere un utente utilizzando la CLI

- Utilizzate il comando [create-acl per creare un ACL](#).

Per Linux, macOS o Unix:

```
aws memorydb create-acl \  
  --acl-name "new-acl-1" \  
  --tags "tag-key=tag-value" \  
  --output text
```

```
--user-names "user-name-1" "user-name-2"
```

Per Windows:

```
aws memorydb create-acl ^  
--acl-name "new-acl-1" ^  
--user-names "user-name-1" "user-name-2"
```

Modifica di una lista di controllo degli accessi (ACL) (console)

Per modificare un elenco di controllo degli accessi utilizzando la console

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Nel riquadro di navigazione a sinistra, scegli Access control lists (ACL).
3. Scegli l'ACL che desideri modificare, quindi scegli Modifica
4. Nella pagina Modifica, in Utenti selezionati, esegui una delle seguenti operazioni:
 - a. Crea un nuovo utente scegliendo Crea utente da aggiungere all'ACL.
 - b. Aggiungi o rimuovi utenti scegliendo Gestisci, quindi selezionando o deselegionando gli utenti dalla finestra di dialogo Gestisci utenti e quindi selezionando Scegli.
5. Nella pagina Crea lista di controllo degli accessi (ACL), inserisci un nome ACL.

I vincoli di denominazione dei cluster sono i seguenti:

- Devono contenere da 1 a 40 caratteri alfanumerici o trattini.
 - Devono iniziare con una lettera.
 - Non possono contenere due trattini consecutivi.
 - Non possono terminare con un trattino.
6. In Utenti selezionati, effettuate una delle seguenti operazioni:
 - a. Crea un nuovo utente selezionando Crea utente
 - b. Aggiungi utenti scegliendo Gestisci, quindi selezionando gli utenti dalla finestra di dialogo Gestisci utenti e quindi selezionando Scegli.
 7. Scegli Modifica per salvare le modifiche o Annulla per eliminarle.

Modifica di una lista di controllo degli accessi (ACL) utilizzando il AWS CLI

Per modificare un ACL aggiungendo nuovi utenti o rimuovendo i membri correnti utilizzando la CLI

- Utilizzate il comando [update-acl per modificare un ACL](#).

Per 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
```

Per 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

Tutte le connessioni aperte appartenenti a un utente rimosso da un ACL vengono terminate con questo comando.

Visualizzazione dei dettagli dell'Access Control List (ACL) (Console)

Per visualizzare i dettagli ACL sulla console

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. Nel riquadro di navigazione a sinistra, scegli Access control lists (ACL).
3. Scegli l'ACL sotto il nome ACL o usa la casella di ricerca per trovare l'ACL.
4. In Utenti puoi visualizzare l'elenco degli utenti associati all'ACL.
5. In Cluster associati è possibile esaminare il cluster a cui appartiene l'ACL.
6. In Tag è possibile esaminare tutti i tag associati all'ACL.

Visualizzazione degli elenchi di controllo degli accessi (ACL) utilizzando AWS CLI

Utilizzate il comando [describe-acls](#) per visualizzare i dettagli di un ACL.

```
aws memorydb describe-acls \  
  --acl-name test-group
```

Eliminazione di un Access Control List (ACL) (console)

Per eliminare gli elenchi di controllo degli accessi utilizzando la console

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. Nel riquadro di navigazione a sinistra, scegli Access control lists (ACL).
3. Scegli l'ACL che desideri modificare, quindi scegli Elimina
4. Nella pagina Elimina, inserisci delete e la casella di conferma e scegli Elimina o Annulla per evitare di eliminare l'ACL.

L'ACL stesso, non gli utenti che appartengono al gruppo, viene eliminato.

Eliminazione di una lista di controllo degli accessi (ACL) utilizzando AWS CLI

Per eliminare un ACL utilizzando la CLI

- Utilizzate il comando [delete-acl per eliminare un ACL](#).

Per Linux, macOS o Unix:

```
aws memorydb delete-acl \  
  --acl-name
```

Per Windows:

```
aws memorydb delete-acl ^ \  
  --acl-name
```

Gli esempi precedenti restituiscono la risposta seguente.

```
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"
}
```

Assegnazione delle liste di controllo degli accessi ai cluster

Dopo aver creato un ACL e aggiunto gli utenti, il passaggio finale dell'implementazione degli ACL consiste nell'assegnare l'ACL a un cluster.

Assegnazione degli elenchi di controllo degli accessi ai cluster mediante la console

Per aggiungere un ACL a un cluster utilizzando il AWS Management Console, vedere. [Creazione di un cluster MemoryDB](#)

Assegnazione delle liste di controllo degli accessi ai cluster Utilizzando il AWS CLI

La seguente AWS CLI operazione crea un cluster con la crittografia in transito (TLS) abilitata e il `acl-name` parametro con il valore. *my-acl-name* Sostituisci il gruppo di sottoreti `subnet-group` con uno esistente.

Parametri chiave

- **--engine-version**— Deve essere 6.2.
- **--tls-enabled**— Utilizzato per l'autenticazione e per associare un ACL.
- **--acl-name**— Questo valore fornisce elenchi di controllo degli accessi composti da utenti con autorizzazioni di accesso specifiche per il cluster.

Per Linux, macOS o Unix:

```
aws memorydb create-cluster \  
  --cluster-name "new-cluster" \  
  --acl-name "my-acl-name" \  
  --subnet-group "subnet-group" \  
  --engine-version "6.2" \  
  --tls-enabled
```

```
--description "new-cluster" \  
--engine-version "6.2" \  
--node-type db.r6g.large \  
--tls-enabled \  
--acl-name "new-acl-1" \  
--subnet-group-name "subnet-group"
```

Per 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 seguente AWS CLI operazione modifica un cluster con la crittografia in transito (TLS) abilitata e il `acl-name` parametro con il valore `new-acl-2`

Per Linux, macOS o Unix:

```
aws memorydb update-cluster \  
  --cluster-name cluster-1 \  
  --acl-name "new-acl-2"
```

Per Windows:

```
aws memorydb update-cluster ^  
  --cluster-name cluster-1 ^  
  --acl-name "new-acl-2"
```

Autenticazione con IAM

Argomenti

- [Panoramica](#)
- [Restrizioni](#)

- [Installazione](#)
- [Connessione](#)

Panoramica

Con l'autenticazione IAM puoi autenticare una connessione a MemoryDB utilizzando identità AWS IAM, quando il cluster è configurato per utilizzare Redis versione 7 o successive. Ciò consente di consolidare il modello di sicurezza e semplificare molte attività di sicurezza amministrative. Con l'autenticazione IAM puoi configurare il controllo granulare degli accessi per ogni singolo cluster MemoryDB e utente di MemoryDB e seguire i principi di autorizzazioni del privilegio minimo. Autenticazione IAM per MemoryDB Redis funziona fornendo un token di autenticazione IAM di breve durata anziché una password utente MemoryDB di lunga durata nel comando Redis o. AUTH HELLO Per ulteriori informazioni sul token di autenticazione IAM, consulta il [processo di firma Signature Version 4](#) nella Guida di riferimento generale di AWS e l'esempio di codice riportato di seguito.

Puoi utilizzare le identità IAM e le policy associate per limitare ulteriormente l'accesso a Redis. Puoi anche concedere l'accesso agli utenti dai loro provider di identità federate

Per utilizzare AWS IAM con MemoryDB, devi prima creare un utente MemoryDB con la modalità di autenticazione impostata su IAM, quindi puoi creare o riutilizzare un'identità IAM. L'identità IAM richiede una policy associata per concedere l'accesso al cluster MemoryDB e all'utente MemoryDB. Una volta configurato, puoi creare un token di autenticazione IAM utilizzando le credenziali AWS dell'utente o del ruolo IAM. Infine, è necessario fornire il token di autenticazione IAM di breve durata come una password nel client Redis durante la connessione al nodo cluster MemoryDB. Un client Redis con supporto per il provider di credenziali può generare automaticamente le credenziali temporanee per ogni nuova connessione. MemoryDB eseguirà l'autenticazione IAM per le richieste di connessione degli utenti di MemoryDB attivate da IAM e convaliderà le richieste di connessione con IAM.

Restrizioni

Durante l'utilizzo dell'autenticazione IAM, valgono le seguenti limitazioni:

- L'autenticazione IAM è disponibile quando si utilizza la versione 7.0 o successive del motore Redis.
- Il token di autenticazione IAM è valido per 15 minuti. Per connessioni di lunga durata, consigliamo di utilizzare un client Redis che supporti l'interfaccia del provider di credenziali.

- Una connessione autenticata da IAM a MemoryDB verrà disconnessa dopo 12 ore. La connessione può essere prolungata per 12 ore inviando un comando AUTH o HELLO con un nuovo token di autenticazione IAM.
- L'autenticazione IAM non è supportata nei comandi MULTI EXEC.
- Attualmente, l'autenticazione IAM non supporta nessuna delle chiavi di contesto delle condizioni globali. Per ulteriori informazioni sulle chiavi di contesto delle condizioni globali, consulta [Chiavi di contesto delle condizioni AWS globali](#) nella Guida per l'utente IAM.

Installazione

Per impostare l'autenticazione IAM:

1. Creazione di un cluster

```
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. Crea un documento della policy di attendibilità IAM per il ruolo, come mostrato di seguito, che consenta all'account di assumere il nuovo ruolo. Salva la policy in un file denominato trust-policy.json.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": { "AWS": "arn:aws:iam::123456789012:root" },  
      "Action": "sts:AssumeRole"  
    }  
  ]  
}
```

3. Crea un documento della policy IAM, come mostrato di seguito. Salva la policy in un file denominato policy.json.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": { "AWS": "arn:aws:iam::123456789012:root" },  
      "Action": "memorydb:*"  
    }  
  ]  
}
```

```
{
  "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. Crea un ruolo IAM.

```
aws iam create-role \
  --role-name "memorydb-iam-auth-app" \
  --assume-role-policy-document file://trust-policy.json
```

5. Creare la policy IAM.

```
aws iam create-policy \
  --policy-name "memorydb-allow-all" \
  --policy-document file://policy.json
```

6. Allegare la policy IAM al ruolo.

```
aws iam attach-role-policy \
  --role-name "memorydb-iam-auth-app" \
  --policy-arn "arn:aws:iam::123456789012:policy/memorydb-allow-all"
```

7. Crea un nuovo utente attivato da IAM.

```
aws memorydb create-user \
  --user-name iam-user-01 \
  --authentication-mode Type=iam \
  --access-string "on ~* +@all"
```

8. Crea un ACL e collega l'utente.

```
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
```

Connessione

Connetti con token come password

È innanzitutto necessario generare il token di autenticazione IAM di breve durata utilizzando una [richiesta prefirmata AWS SigV4](#). Dopodiché, occorre fornire il token di autenticazione IAM come una password durante la connessione a un cluster MemoryDB, come mostrato nell'esempio riportato di seguito.

```
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();
```

Di seguito è riportata la definizione per 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());
}
}

```

Connetti con provider di credenziali

Il codice seguente mostra come eseguire l'autenticazione con MemoryDB utilizzando il provider di credenziali dell'autenticazione 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();
```

Di seguito è riportato un esempio di client cluster Redis Lettuce che racchiude l'IAM AuthTokenRequest in un provider di credenziali per generare automaticamente credenziali temporanee quando richiesto.

```
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());  
}
```

Gestione delle identità e degli accessi in MemoryDB per Redis

AWS Identity and Access Management (IAM) è un programma Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse MemoryDB. IAM è uno strumento Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso con policy](#)
- [Come funziona MemoryDB for Redis con IAM](#)
- [Esempi di policy basate sull'identità per MemoryDB for Redis](#)
- [Risoluzione dei problemi relativi all'identità e all'accesso a MemoryDB per Redis](#)
- [Controllo accessi](#)
- [Panoramica della gestione delle autorizzazioni di accesso alle risorse di MemoryDB](#)

Destinatari

Il modo in cui si utilizza AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in MemoryDB.

Utente del servizio: se utilizzi il servizio MemoryDB per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più funzionalità di MemoryDB per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La

comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità di MemoryDB, consulta. [Risoluzione dei problemi relativi all'identità e all'accesso a MemoryDB per Redis](#)

Amministratore del servizio: se sei responsabile delle risorse di MemoryDB presso la tua azienda, probabilmente hai pieno accesso a MemoryDB. È tuo compito determinare a quali funzionalità e risorse di MemoryDB devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per saperne di più su come la tua azienda può utilizzare IAM con MemoryDB, consulta. [Come funziona MemoryDB for Redis con IAM](#)

Amministratore IAM: se sei un amministratore IAM, potresti voler conoscere i dettagli su come scrivere politiche per gestire l'accesso a MemoryDB. Per visualizzare esempi di policy basate sull'identità di MemoryDB che puoi utilizzare in IAM, consulta. [Esempi di policy basate sull'identità per MemoryDB for Redis](#)

Autenticazione con identità

L'autenticazione è il modo in cui accedi utilizzando le tue credenziali di identità. AWS Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi [Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente](#).

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta [Signing AWS API request](#) nella IAM User Guide.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta [Autenticazione a più fattori](#) nella Guida per l'utente di AWS IAM Identity Center e [Utilizzo dell'autenticazione a più fattori \(MFA\) in AWS](#) nella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente IAM.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni su IAM Identity Center, consulta [Cos'è IAM Identity Center?](#) nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un [utente IAM](#) è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, se si hanno casi d'uso specifici che richiedono credenziali a lungo termine con

utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina [Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine](#) nella Guida per l'utente IAM.

Un [gruppo IAM](#) è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile avere un gruppo denominato IAMAdmins e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta [Quando creare un utente IAM \(invece di un ruolo\)](#) nella Guida per l'utente IAM.

Ruoli IAM

Un [ruolo IAM](#) è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Puoi assumere temporaneamente un ruolo IAM in AWS Management Console [cambiando ruolo](#). Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta [Utilizzo di ruoli IAM](#) nella Guida per l'utente IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- **Accesso utente federato:** per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta [Creazione di un ruolo per un provider di identità di terza parte](#) nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta [Set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center .
- **Autorizzazioni utente IAM temporanee:** un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.

- **Accesso multi-account:** è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per conoscere la differenza tra ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta [Cross Account Resource Access in IAM nella IAM User Guide](#).
- **Accesso tra servizi:** alcuni Servizi AWS utilizzano funzionalità in altri. Servizi AWS Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
- **Sessioni di accesso diretto (FAS):** quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un preside. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina [Forward access sessions](#).
- **Ruolo di servizio:** un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire azioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS](#) nella Guida per l'utente IAM.
- **Ruolo collegato al servizio:** un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- **Applicazioni in esecuzione su Amazon EC2:** puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che AWS CLI effettuano richieste API. AWS. Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un AWS ruolo a un'istanza EC2 e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori

informazioni, consulta [Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2](#) nella Guida per l'utente IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta [Quando creare un ruolo IAM \(invece di un utente\)](#) nella Guida per l'utente IAM.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall' AWS CLI o dall' AWS API.

Policy basate su identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Creazione di policy IAM](#) nella Guida per l'utente IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche

gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta [Scelta fra policy gestite e policy inline](#) nella Guida per l'utente IAM.

Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Liste di controllo degli accessi (ACL)

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano gli ACL. AWS WAF Per maggiori informazioni sulle ACL, consulta [Panoramica delle liste di controllo degli accessi \(ACL\)](#) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- **Limiti delle autorizzazioni:** un limite delle autorizzazioni è una funzionalità avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di

queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente IAM.

- **Politiche di controllo dei servizi (SCP):** le SCP sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in AWS Organizations. AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna. Utente root dell'account AWS Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations .
- **Policy di sessione:** le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta [Policy di sessione](#) nella Guida per l'utente IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle policy](#) nella IAM User Guide.

Come funziona MemoryDB for Redis con IAM

Prima di utilizzare IAM per gestire l'accesso a MemoryDB, scopri quali funzionalità IAM sono disponibili per l'uso con MemoryDB.

Funzionalità IAM che puoi utilizzare con MemoryDB for Redis

Funzionalità IAM	Supporto per MemoryDB
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì

Funzionalità IAM	Supporto per MemoryDB
Risorse relative alle policy	Sì
Chiavi di condizione delle policy	Sì
Liste di controllo degli accessi (ACL)	Sì
ABAC (tag nelle policy)	Sì
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	Sì

Per avere una visione di alto livello di come MemoryDB e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta [AWS i servizi che funzionano con IAM nella IAM User Guide](#).

Politiche basate sull'identità per MemoryDB

Supporta le policy basate su identità	Sì
---------------------------------------	----

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Creazione di policy IAM](#) nella Guida per l'utente IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta [Guida di riferimento agli elementi delle policy JSON IAM](#) nella Guida per l'utente di IAM.

Esempi di policy basate sull'identità per MemoryDB

Per visualizzare esempi di politiche basate sull'identità di MemoryDB, vedere. [Esempi di policy basate sull'identità per MemoryDB for Redis](#)

Politiche basate sulle risorse all'interno di MemoryDB

Supporta le policy basate su risorse	No
--------------------------------------	----

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta [Cross Account Resource Access in IAM](#) nella IAM User Guide.

Azioni politiche per MemoryDB

Supporta le operazioni di policy	Sì
----------------------------------	----

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Actions` di una policy JSON descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le azioni di sola autorizzazione che non

hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco delle azioni di MemoryDB, consulta [Azioni definite da MemoryDB per Redis](#) nel Service Authorization Reference.

Le azioni politiche in MemoryDB utilizzano il seguente prefisso prima dell'azione:

```
MemoryDB
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [  
  "MemoryDB:action1",  
  "MemoryDB:action2"  
]
```

È possibile specificare più azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola Describe, includi la seguente azione:

```
"Action": "MemoryDB:Describe*"
```

Per visualizzare esempi di politiche basate sull'identità di MemoryDB, vedere. [Esempi di policy basate sull'identità per MemoryDB for Redis](#)

Risorse politiche per MemoryDB

Supporta le risorse di policy	Si
-------------------------------	----

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`.

Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Per visualizzare un elenco dei tipi di risorse di MemoryDB e dei relativi ARN, consulta [Resources Defined by MemoryDB for Redis](#) nel Service Authorization Reference. Per sapere con quali azioni è possibile specificare l'ARN di ciascuna risorsa, vedere [Azioni definite da MemoryDB per Redis](#).

Per visualizzare esempi di politiche basate sull'identità di MemoryDB, vedi [Esempi di policy basate sull'identità per MemoryDB for Redis](#)

Chiavi relative alle condizioni dei criteri per MemoryDB

Supporta le chiavi di condizione delle policy specifiche del servizio	Sì
---	----

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento Condition(o blocco Condition) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento Conditionè facoltativo. Puoi compilare espressioni condizionali che utilizzano [operatori di condizione](#), ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi Conditionin un'istruzione o più chiavi in un singolo elemento Condition, questi vengono valutati da AWS utilizzando un'operazione ANDlogica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. OR Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome

utente IAM. Per ulteriori informazioni, consulta [Elementi delle policy IAM: variabili e tag](#) nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali nella Guida](#) per l'utente IAM.

Per visualizzare esempi di politiche basate sull'identità di MemoryDB, consulta [Esempi di policy basate sull'identità per MemoryDB for Redis](#)

Utilizzo delle chiavi di condizione

Puoi specificare le condizioni che determinano il modo in cui una policy IAM viene applicata. In MemoryDB, è possibile utilizzare l'Conditionelemento di una policy JSON per confrontare le chiavi nel contesto della richiesta con i valori chiave specificati nella policy. Per ulteriori informazioni, consulta [elementi della policy IAM JSON: condizione](#).

Per visualizzare un elenco di chiavi di condizione di MemoryDB, consulta Condition Keys [for MemoryDB for Redis](#) nel Service Authorization Reference.

Per un elenco delle chiavi di condizione globali, consulta [Chiavi di contesto delle condizioni globali AWS](#).

Specifiche delle condizioni: Uso delle chiavi di condizione

Per implementare un controllo granulare, puoi scrivere una policy di autorizzazioni IAM che specifichi le condizioni per controllare un set di singoli parametri su determinate richieste. Puoi quindi applicare la policy agli utenti, ai gruppi o ai ruoli IAM che crei utilizzando la console IAM.

Per applicare una condizione, aggiungere le informazioni sulla condizione all'istruzione della policy IAM. Ad esempio, per impedire la creazione di qualsiasi cluster MemoryDB con TLS disabilitato, puoi specificare la seguente condizione nella tua dichiarazione politica.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "memorydb:CreateCluster"
      ],
      "Resource": [
```

```

    "*"
  ],
  "Condition": {
    "Bool": {
      "memorydb:TLSEnabled": "false"
    }
  }
}
]
}

```

Per ulteriori informazioni sull'etichettatura, vedere. [Tagging delle risorse MemoryDB](#)

Per ulteriori informazioni sull'utilizzo degli operatori delle condizioni di policy, consulta [Autorizzazioni API MemoryDB: riferimento ad azioni, risorse e condizioni](#).

Policy di esempio: Utilizzo di condizioni per il controllo granulare dei parametri

Questa sezione mostra esempi di policy per l'implementazione di un controllo granulare degli accessi sui parametri MemoryDB elencati in precedenza.

1. memoryDB:TLSEnabled — Specificate che i cluster verranno creati solo con TLS abilitato.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "memorydb:CreateCluster"
      ],
      "Resource": [
        "arn:aws:memorydb:*:*:parametergroup/*",
        "arn:aws:memorydb:*:*:subnetgroup/*",
        "arn:aws:memorydb:*:*:acl/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "memorydb:CreateCluster"
      ],
      "Resource": [

```



```

        "*"
      ],
      "Condition": {
        "Bool": {
          "memorydb:TLSEnabled": "true"
        }
      }
    }
  ]
}

```

2. `memorydb:UserAuthenticationMode`: — Specificate che gli utenti possono essere creati con una modalità di autenticazione di tipo specifico (IAM per esempio).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "memorydb:Createuser"
      ],
      "Resource": [
        "arn:aws:memorydb:*:*:user/*"
      ],
      "Condition": {
        "StringEquals": {
          "memorydb:UserAuthenticationMode": "iam"
        }
      }
    }
  ]
}

```

Nei casi in cui si impostano politiche basate su «Deny», si consiglia di utilizzare l'[StringEqualsIgnoreCase](#) operatore per evitare tutte le chiamate con un tipo di modalità di autenticazione utente specifico, indipendentemente dal caso.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {

```

```
"Effect": "Deny",
"Action": [
  "memorydb:CreateUser"
],
"Resource": "*",
"Condition": {
  "StringEqualsIgnoreCase": {
    "memorydb:UserAuthenticationMode": "password"
  }
}
]
```

Liste di controllo degli accessi (ACL) in MemoryDB

Supporta le ACL	Si
-----------------	----

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Controllo degli accessi basato sugli attributi (ABAC) con MemoryDB

Supporta ABAC (tag nelle policy)	Si
----------------------------------	----

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In, questi attributi sono chiamati AWS tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'[elemento condizione](#) di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta [Che cos'è ABAC?](#) nella Guida per l'utente IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta [Utilizzo del controllo degli accessi basato su attributi \(ABAC\)](#) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con MemoryDB

Supporta le credenziali temporanee	Sì
------------------------------------	----

Alcune Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla [Servizi AWS compatibilità con IAM nella IAM User Guide](#).

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedi AWS utilizzando il link Single Sign-On (SSO) della tua azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta [Cambio di un ruolo \(console\)](#) nella Guida per l'utente IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API o AWS CLI. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta [Credenziali di sicurezza provvisorie in IAM](#).

Autorizzazioni principali multiservizio per MemoryDB

Supporta l'inoltro delle sessioni di accesso (FAS)	Sì
--	----

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina [Forward access sessions](#).

Ruoli di servizio per MemoryDB

Supporta i ruoli di servizio	Sì
------------------------------	----

Un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS](#) nella Guida per l'utente IAM.

Warning

La modifica delle autorizzazioni per un ruolo di servizio potrebbe interrompere la funzionalità di MemoryDB. Modifica i ruoli di servizio solo quando MemoryDB fornisce indicazioni in tal senso.

Ruoli collegati ai servizi per MemoryDB

Supporta i ruoli collegati ai servizi	Sì
---------------------------------------	----

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi, consulta [Servizi AWS supportati da IAM](#). Trova un servizio nella tabella che include un Yes nella colonna Service-linked

ruolo (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Esempi di policy basate sull'identità per MemoryDB for Redis

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse MemoryDB. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS l'API. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta [Creazione di policy IAM](#) nella Guida per l'utente di IAM.

Per i dettagli sulle azioni e sui tipi di risorse definiti da MemoryDB, incluso il formato degli ARN per ciascun tipo di risorsa, consulta [Actions, Resources and Condition Keys for MemoryDB for Redis](#) nel Service Authorization Reference.

Argomenti

- [Best practice per le policy](#)
- [Utilizzo della console MemoryDB](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare le risorse MemoryDB nel tuo account. Queste azioni possono comportare costi aggiuntivi per l'Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni. AWS Sono disponibili nel tuo Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni

che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta [Policy e autorizzazioni in IAM](#) nella Guida per l'utente IAM.

- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse puoi aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta [Convalida delle policy per IAM Access Analyzer](#) nella Guida per l'utente IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta [Configurazione dell'accesso alle API protetto con MFA](#) nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta [Best practice di sicurezza in IAM](#) nella Guida per l'utente di IAM.

Utilizzo della console MemoryDB

Per accedere alla console MemoryDB for Redis, è necessario disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse MemoryDB presenti nel tuo Account AWS. Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso o l' AWS CLI API. AWS Al contrario, concedi l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano ancora utilizzare la console MemoryDB, collega anche MemoryDB ConsoleAccess o la policy ReadOnly AWS gestita alle entità. Per ulteriori informazioni, consulta [Aggiunta di autorizzazioni a un utente](#) nella Guida per l'utente IAM.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

```
}
```

Risoluzione dei problemi relativi all'identità e all'accesso a MemoryDB per Redis

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con MemoryDB e IAM.

Argomenti

- [Non sono autorizzato a eseguire un'azione in MemoryDB](#)
- [Non sono autorizzato a eseguire iam: PassRole](#)
- [Voglio consentire a persone esterne al mio AWS account di accedere alle mie risorse MemoryDB](#)

Non sono autorizzato a eseguire un'azione in MemoryDB

Se ti AWS Management Console dice che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è la persona da cui si sono ricevuti il nome utente e la password.

Il seguente esempio di errore si verifica quando l'utente mateojackson prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *my-example-widget* fittizia, ma non dispone di autorizzazioni MemoryDB: *GetWidget* fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
MemoryDB: GetWidget on resource: my-example-widget
```

In questo caso, Mateo richiede al suo amministratore di aggiornare le policy per poter accedere alla risorsa *my-example-widget* utilizzando l'operazione MemoryDB: *GetWidget*.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un errore che indica che non sei autorizzato a eseguire l'iam:PassRole azione, le tue politiche devono essere aggiornate per consentirti di passare un ruolo a MemoryDB.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

Il seguente errore di esempio si verifica quando un utente IAM denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in MemoryDB. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne al mio AWS account di accedere alle mie risorse MemoryDB

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per servizi che supportano policy basate su risorse o liste di controllo degli accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se MemoryDB supporta queste funzionalità, consulta [Come funziona MemoryDB for Redis con IAM](#)
- Per scoprire come fornire l'accesso alle risorse di tua proprietà, consulta [Fornire l'accesso a un utente IAM di un altro Account AWS utente di tua proprietà nella IAM User Guide](#). Account AWS
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta [Fornire l'accesso a soggetti Account AWS di proprietà di terze parti](#) nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta [Fornire l'accesso a utenti autenticati esternamente \(Federazione delle identità\)](#) nella Guida per l'utente IAM.
- Per scoprire la differenza tra l'utilizzo di ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta [Cross Account Resource Access in IAM nella IAM User Guide](#).

Controllo accessi

Puoi avere credenziali valide per autenticare le tue richieste, ma a meno che tu non disponga delle autorizzazioni non puoi creare o accedere alle risorse di MemoryDB for Redis. Ad esempio, è necessario disporre delle autorizzazioni per creare un cluster MemoryDB.

Le sezioni seguenti descrivono come gestire le autorizzazioni per MemoryDB for Redis. Consigliamo di leggere prima la panoramica.

- [Panoramica della gestione delle autorizzazioni di accesso alle risorse di MemoryDB](#)
- [Utilizzo di politiche basate sull'identità \(politiche IAM\) per MemoryDB for Redis](#)

Panoramica della gestione delle autorizzazioni di accesso alle risorse di MemoryDB

Ogni AWS risorsa è di proprietà di un AWS account e le autorizzazioni per creare o accedere a una risorsa sono regolate dalle politiche di autorizzazione. Un amministratore dell'account è in grado di collegare le policy relative alle autorizzazioni alle identità IAM (ovvero utenti, gruppi e ruoli). Inoltre, MemoryDB for Redis supporta anche l'associazione di politiche di autorizzazione alle risorse.

Note

Un amministratore account (o un utente amministratore) è un utente con privilegi di amministratore. Per ulteriori informazioni, consulta [Best practice IAM](#) nella Guida per l'utente di IAM.

Per fornire l'accesso, aggiungi autorizzazioni ai tuoi utenti, gruppi o ruoli:

- Utenti e gruppi in: AWS IAM Identity Center

Crea un set di autorizzazioni. Segui le istruzioni riportate nella pagina [Create a permission set](#) (Creazione di un set di autorizzazioni) nella Guida per l'utente di AWS IAM Identity Center .

- Utenti gestiti in IAM tramite un provider di identità:

Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina [Creating a role for a third-party identity provider \(federation\)](#) (Creazione di un ruolo per un provider di identità di terze parti [federazione]) nella Guida per l'utente di IAM.

- Utenti IAM:

- Crea un ruolo che l'utente possa assumere. Per istruzioni, consulta la pagina [Creating a role for an IAM user](#) (Creazione di un ruolo per un utente IAM) nella Guida per l'utente di IAM.
- (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate nella pagina [Aggiunta di autorizzazioni a un utente \(console\)](#) nella Guida per l'utente di IAM.

Argomenti

- [MemoryDB per risorse e operazioni Redis](#)
- [Informazioni sulla proprietà delle risorse](#)

- [Gestione dell'accesso alle risorse](#)
- [Utilizzo di politiche basate sull'identità \(politiche IAM\) per MemoryDB for Redis](#)
- [Autorizzazioni a livello di risorsa](#)
- [Utilizzo di ruoli collegati ai servizi per Amazon MemoryDB for Redis](#)
- [AWSpolicy gestite per MemoryDB for Redis](#)
- [Autorizzazioni API MemoryDB: riferimento ad azioni, risorse e condizioni](#)

MemoryDB per risorse e operazioni Redis

In MemoryDB for Redis, la risorsa principale è un cluster.

Alle risorse sono associati Amazon Resource Name (ARN) univoci, come illustrato di seguito.

Note

Affinché le autorizzazioni a livello di risorsa siano efficaci, il nome della risorsa nella stringa ARN deve essere minuscolo.

Tipo di risorsa	Formato ARN
Utente	<i>arn:aws:memorydb: us-east-1:123456789012: utente/utente1</i>
Elenco di controllo degli accessi (ACL)	<i>arn:aws:memorydb: us-east-1:123456789012:acl/myacl</i>
Cluster	<i>arn: aws:memorydb: us-east-1:123456789012: cluster/my-cluster</i>
Snapshot	<i>arn:aws:memorydb: us-east-1:123456789012: snapshot/my-snapshot</i>

Tipo di risorsa	Formato ARN
Gruppo di parametri	<i>arn: aws:memorydb: us-east-1:123456789012: parametergroup/ my-parameter-group</i>
Subnet group (Gruppo di sottoreti)	<i>arn:aws:memorydb: us-east-1:123456789012: subnetgroup/ my-subnet-group</i>

MemoryDB fornisce una serie di operazioni per lavorare con le risorse di MemoryDB. [Per un elenco delle operazioni disponibili, consulta MemoryDB for Redis Actions.](#)

Informazioni sulla proprietà delle risorse

Il proprietario della risorsa è l' AWS account che ha creato la risorsa. In altre parole, il proprietario della risorsa è l' AWS account dell'entità principale che autentica la richiesta che crea la risorsa. Un'entità principale può essere l'account root, un utente IAM o un ruolo IAM. Negli esempi seguenti viene illustrato il funzionamento:

- Supponiamo di utilizzare le credenziali dell'account root del proprio AWS account per creare un cluster. In questo caso, il tuo AWS account è il proprietario della risorsa. In MemoryDB, la risorsa è il cluster.
- Supponiamo di creare un utente IAM nel tuo AWS account e di concedere a quell'utente le autorizzazioni per creare un cluster. In questo caso, l'utente può creare un cluster. Tuttavia, l' AWS account a cui appartiene l'utente è proprietario della risorsa del cluster.
- Supponiamo che tu crei un ruolo IAM nel tuo AWS account con le autorizzazioni per creare un cluster. In questo caso, chiunque possa assumere il ruolo può creare un cluster. Il tuo AWS account, a cui appartiene il ruolo, possiede la risorsa del cluster.

Gestione dell'accesso alle risorse

La policy delle autorizzazioni descrive chi ha accesso a cosa. Nella sezione seguente vengono descritte le opzioni disponibili per la creazione di policy relative alle autorizzazioni.

Note

Questa sezione illustra l'utilizzo di IAM nel contesto di MemoryDB per Redis. Non vengono fornite informazioni dettagliate sul servizio IAM. Per la documentazione di IAM completa, consulta [Che cos'è IAM?](#) nella Guida per l'utente di IAM. Per informazioni sulla sintassi delle policy IAM e le rispettive descrizioni, consulta [Riferimento alle policy IAM di AWS](#) nella Guida per l'utente di IAM.

Le policy collegate a un'identità IAM vengono definite policy basate su identità (policy IAM). Le policy collegate a una risorsa vengono definite policy basate sulle risorse.

Argomenti

- [Policy basate su identità \(policy IAM\)](#)
- [Specifica degli elementi delle policy: operazioni, effetti, risorse ed entità](#)
- [Specifica delle condizioni in una policy](#)

Policy basate su identità (policy IAM)

Puoi collegare le policy alle identità IAM. Ad esempio, puoi eseguire le operazioni seguenti:

- Collegare una policy di autorizzazione a un utente o a un gruppo nell'account – Per assegnare le autorizzazioni un amministratore di account può utilizzare una policy di autorizzazione associata a un utente specifico. In questo caso, l'utente è autorizzato a creare una risorsa MemoryDB, come un cluster, un gruppo di parametri o un gruppo di sicurezza.
- Collega una policy di autorizzazione a un ruolo (assegnazione di autorizzazioni tra account): per concedere autorizzazioni tra più account, è possibile collegare una policy di autorizzazione basata su identità a un ruolo IAM. Ad esempio, l'amministratore dell'account A può creare un ruolo per concedere autorizzazioni su più account a un altro account (ad esempio, l' AWS account B) o a un servizio nel modo seguente: AWS
 1. L'amministratore dell'account A crea un ruolo IAM e attribuisce una policy di autorizzazione al ruolo che concede le autorizzazioni sulle risorse per l'account A.
 2. L'amministratore dell'account A attribuisce una policy di attendibilità al ruolo, identificando l'account B come il principale per tale ruolo.
 3. L'amministratore dell'Account B può quindi delegare le autorizzazioni per assumere il ruolo a qualsiasi utente dell'Account B. In questo modo gli utenti dell'Account B possono creare o

accedere alle risorse dell'Account A. In alcuni casi, potresti voler concedere a un AWS servizio le autorizzazioni per assumere il ruolo. Per supportare tale approccio, l'entità principale nella policy di trust può anche essere un'entità principale di un servizio AWS .

Per ulteriori informazioni sull'uso di IAM per delegare le autorizzazioni, consulta [Access Management](#) nella IAM User Guide (Guida per l'utente di IAM).

Di seguito è riportato un esempio di politica che consente a un utente di eseguire l'DescribeClustersazione per il tuo account. AWS MemoryDB supporta anche l'identificazione di risorse specifiche utilizzando le risorse ARN per le azioni API. (questo approccio è anche noto come autorizzazioni a livello di risorsa).

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "DescribeClusters",
    "Effect": "Allow",
    "Action": [
      "memorydb:DescribeClusters"],
    "Resource": resource-arn
  ]
}
```

Per ulteriori informazioni sull'utilizzo di politiche basate sull'identità con MemoryDB, vedere. [Utilizzo di politiche basate sull'identità \(politiche IAM\) per MemoryDB for Redis](#) Per ulteriori informazioni su utenti, gruppi, ruoli e autorizzazioni, consulta [Identità \(utenti, gruppi e ruoli\)](#) nella Guida per l'utente di IAM.

Specifica degli elementi delle policy: operazioni, effetti, risorse ed entità

[Per ogni risorsa MemoryDB for Redis \(vediMemoryDB per risorse e operazioni Redis\), il servizio definisce un insieme di operazioni API \(vedi Azioni\).](#) Per concedere le autorizzazioni per queste operazioni API, MemoryDB definisce una serie di azioni che è possibile specificare in una politica. Ad esempio, per la risorsa del cluster MemoryDB, vengono definite le seguenti azioni:, e. CreateCluster DeleteCluster DescribeClusters L'esecuzione di un'operazione API può richiedere le autorizzazioni per più di un'operazione.

Di seguito sono elencati gli elementi di base di una policy:

- **Risorsa:** in una policy si utilizza il nome della risorsa Amazon (ARN) per identificare la risorsa a cui si applica la policy stessa. Per ulteriori informazioni, consulta [MemoryDB per risorse e operazioni Redis](#).
- **Operazione:** utilizzi le parole chiave per identificare le operazioni sulla risorsa da permettere o rifiutare. Ad esempio, a seconda di quanto specificato `Effect`, `memorydb:CreateCluster` autorizzazione consente o nega all'utente le autorizzazioni per eseguire l'operazione MemoryDB for Redis. `CreateCluster`
- **Effetto:** l'effetto prodotto quando l'utente richiede l'operazione specifica, ovvero un'autorizzazione o un rifiuto. `Use` non concedi esplicitamente (consenti) l'accesso a una risorsa, l'accesso viene implicitamente rifiutato. È anche possibile negare esplicitamente l'accesso a una risorsa. Ad esempio, è possibile eseguire questa operazione per accertarsi che un utente non sia in grado di accedere a una risorsa, anche se l'accesso viene concesso da un'altra policy.
- **Principale:** nelle policy basate su identità (policy IAM), l'utente a cui la policy è collegata è il principale implicito. Per policy basate su risorse, specifichi l'utente, l'account, il servizio o un'altra entità che desideri riceva le autorizzazioni (si applica solo alle policy basate su risorse).

Per ulteriori informazioni sulla sintassi e le descrizioni delle policy IAM, consulta [AWS Riferimento alle policy IAM](#) nella Guida per l'utente di IAM.

Per una tabella che mostra tutte le azioni dell'API MemoryDB for Redis, consulta. [Autorizzazioni API MemoryDB: riferimento ad azioni, risorse e condizioni](#)

Specifiche delle condizioni in una policy

Quando si concedono le autorizzazioni, è possibile utilizzare il linguaggio della policy IAM per specificare le condizioni in base a cui la policy deve essere applicata. Ad esempio, potresti decidere che una policy venga applicata solo dopo una data specifica. Per ulteriori informazioni su come specificare le condizioni in un linguaggio di policy, consulta la sezione [Condizione](#) nella Guida per l'utente di IAM.

Utilizzo di politiche basate sull'identità (politiche IAM) per MemoryDB for Redis

In questo argomento vengono forniti esempi di policy basate su identità in cui un amministratore account può collegare policy di autorizzazione a identità IAM, ovvero utenti, gruppi e ruoli.

Important

Ti consigliamo di leggere prima gli argomenti che spiegano i concetti e le opzioni di base per gestire l'accesso alle risorse di MemoryDB for Redis. Per ulteriori informazioni, consulta la pagina [Panoramica della gestione delle autorizzazioni di accesso alle risorse di MemoryDB](#).

In questa sezione vengono trattati gli argomenti seguenti:

- [Autorizzazioni necessarie per utilizzare la console MemoryDB for Redis](#)
- [AWSPolicy gestite da \(predefinite\) per MemoryDB for Redis](#)
- [Esempi di policy gestite dal cliente](#)

Di seguito viene illustrato un esempio di policy di autorizzazione.

```
{
  "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 policy include due dichiarazioni:

- La prima istruzione concede le autorizzazioni per le azioni di MemoryDB for Redis (`memorydb:CreateCluster`, `memorydb:DescribeClusters`, `andmemorydb:UpdateCluster`) su qualsiasi cluster di proprietà dell'account.
- La seconda istruzione concede le autorizzazioni per l'operazione IAM (`iam:PassRole`) sul nome del ruolo IAM specificato alla fine del valore `Resource`.

La policy non specifica l'elemento `Principal` poiché in una policy basata su identità l'entità che ottiene l'autorizzazione non viene specificata. Quando si collega una policy a un utente, quest'ultimo è l'entità implicita. Quando colleghi una policy di autorizzazioni a un ruolo IAM, il principale identificato nella policy di attendibilità del ruolo ottiene le autorizzazioni.

Per una tabella che mostra tutte le azioni dell'API MemoryDB for Redis e le risorse a cui si applicano, vedi [Autorizzazioni API MemoryDB: riferimento ad azioni, risorse e condizioni](#)

Autorizzazioni necessarie per utilizzare la console MemoryDB for Redis

La tabella di riferimento delle autorizzazioni elenca le operazioni dell'API MemoryDB for Redis e mostra le autorizzazioni richieste per ciascuna operazione. Per ulteriori informazioni sulle operazioni dell'API MemoryDB, vedere [Autorizzazioni API MemoryDB: riferimento ad azioni, risorse e condizioni](#)

Per utilizzare la console MemoryDB for Redis, concedi innanzitutto le autorizzazioni per azioni aggiuntive, come mostrato nella seguente politica di autorizzazione.

```
{
  "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 console MemoryDB necessita di queste autorizzazioni aggiuntive per i seguenti motivi:

- Le autorizzazioni per le azioni MemoryDB consentono alla console di visualizzare le risorse MemoryDB nell'account.
- La console richiede le autorizzazioni per le operazioni ec2 per inviare le query ad Amazon EC2 al fine di poter visualizzare zone di disponibilità, VPC, gruppi di sicurezza e attributi dell'account.
- Le cloudwatch autorizzazioni per le azioni consentono alla console di recuperare CloudWatch metriche e allarmi di Amazon e di visualizzarli nella console.
- Le autorizzazioni per le operazioni sns consentono alla console di recuperare argomenti e sottoscrizioni di Amazon Simple Notification Service (Amazon SNS) e mostrarli.

Esempi di policy gestite dal cliente

Se non si utilizza una policy predefinita e si sceglie di utilizzare una policy gestita in modo personalizzato, assicurarsi di trovarsi in una delle due seguenti situazioni. O si dispone delle autorizzazioni per richiamare `iam:createServiceLinkedRole` (Per ulteriori informazioni, consulta [Esempio 4: consenti a un utente di chiamare l'API IAM CreateServiceLinkedRole](#)). Oppure avresti dovuto creare un ruolo collegato al servizio MemoryDB.

Se combinate con le autorizzazioni minime necessarie per utilizzare la console MemoryDB for Redis, le politiche di esempio in questa sezione concedono autorizzazioni aggiuntive. Gli esempi sono rilevanti anche per gli SDK e per AWS CLI. Per ulteriori informazioni sulle autorizzazioni necessarie per utilizzare la console MemoryDB, consulta [Autorizzazioni necessarie per utilizzare la console MemoryDB for Redis](#)

Per istruzioni su come impostare gruppi e utenti IAM, consulta [Creazione del primo utente e gruppo di amministratori IAM](#) nella Guida per l'utente di IAM.

Important

Testa sempre in modo approfondito le Policy IAM prima di avvalertene in fase di produzione. Alcune azioni di MemoryDB che sembrano semplici possono richiedere

altre azioni per supportarle quando si utilizza la console MemoryDB. Ad esempio, `memorydb:CreateCluster` concede le autorizzazioni per creare cluster MemoryDB. Tuttavia, per eseguire questa operazione, la console MemoryDB utilizza una serie di azioni per compilare gli elenchi delle `Describe` console. `List`

Esempi

- [Esempio 1: consentire a un utente l'accesso in sola lettura alle risorse di MemoryDB](#)
- [Esempio 2: consentire a un utente di eseguire attività comuni di amministratore del sistema MemoryDB](#)
- [Esempio 3: consentire a un utente di accedere a tutte le azioni dell'API MemoryDB](#)
- [Esempio 4: consenti a un utente di chiamare l'API IAM `CreateServiceLinkedRole`](#)

Esempio 1: consentire a un utente l'accesso in sola lettura alle risorse di MemoryDB

La seguente politica concede le autorizzazioni per le azioni di MemoryDB che consentono a un utente di elencare le risorse. In genere, si collega questo tipo di policy di autorizzazione a un gruppo di gestori.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "MemDBUnrestricted",
    "Effect": "Allow",
    "Action": [
      "memorydb:Describe*",
      "memorydb:List*"
    ],
    "Resource": "*"
  }
]
```

Esempio 2: consentire a un utente di eseguire attività comuni di amministratore del sistema MemoryDB

Le attività comuni dell'amministratore di sistema includono la modifica di cluster, parametri e gruppi di parametri. Un amministratore di sistema può anche voler ottenere informazioni sugli eventi di MemoryDB. La seguente politica concede a un utente le autorizzazioni per eseguire azioni di

MemoryDB per queste attività comuni dell'amministratore di sistema. In genere, si collega questo tipo di policy di autorizzazione al gruppo degli amministratori di 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": "*"
  }
]
```

Esempio 3: consentire a un utente di accedere a tutte le azioni dell'API MemoryDB

La seguente politica consente a un utente di accedere a tutte le azioni di MemoryDB. Consigliamo di concedere questo tipo di policy di autorizzazione solo a un utente amministratore.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "MDBAllowAll",
    "Effect": "Allow",
    "Action": [
      "memorydb:*"
    ],
    "Resource": "*"
  }
]
```

Esempio 4: consenti a un utente di chiamare l'API IAM CreateServiceLinkedRole

La policy seguente permette a un utente di chiamare l'API IAM `CreateServiceLinkedRole`. Si consiglia di concedere questo tipo di politica di autorizzazione all'utente che richiama operazioni mutative di MemoryDB.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateSLRAllows",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:AWS ServiceName": "memorydb.amazonaws.com"
        }
      }
    }
  ]
}
```

Autorizzazioni a livello di risorsa

È possibile limitare la portata delle autorizzazioni specificando le risorse in una policy IAM. Molte azioni AWS CLI API supportano un tipo di risorsa che varia a seconda del comportamento dell'azione. Ogni dichiarazione di policy IAM concede l'autorizzazione a un'operazione eseguita su una risorsa. Quando l'operazione non agisce su una risorsa designata oppure quando concedi l'autorizzazione per eseguire l'operazione su tutte le risorse, il valore della risorsa nella policy è un carattere jolly (*). Per molte operazioni API è possibile limitare le risorse che un utente può modificare specificando l'Amazon Resource Name (ARN) di una risorsa o un modello ARN che soddisfa più risorse. Per limitare le autorizzazioni in base alla risorsa, specifica la risorsa in base all'ARN.

Formato ARN delle risorse MemoryDB

Note

Affinché le autorizzazioni a livello di risorsa siano efficaci, il nome della risorsa nella stringa ARN deve essere minuscolo.

- *Utente* – `arn:aws:memorydb: us-east- 1:123456789012:user/user1`
- *ACL* – `arn:aws:memorydb: us-east- 1:123456789012:acl/my-acl`
- *Cluster* – `arn:aws:memorydb: us-east- 1:123456789012:cluster/my-cluster`
- *Istantanea* – `arn:aws:memorydb: us-east- 1:123456789012:snapshot/my-snapshot`
- *Gruppo di parametri* – `arn:aws:memorydb: us-east- 1:123456789012:parametergroup/ my-parameter-group`
- *Gruppo di sottoreti* – `arn:aws:memorydb: us-east- 1:123456789012:subnetgroup/ my-subnet-group`

Esempi

- [Esempio 1: consentire a un utente l'accesso completo a tipi di risorse MemoryDB specifici](#)
- [Esempio 2: negare a un utente l'accesso a un cluster.](#)

Esempio 1: consentire a un utente l'accesso completo a tipi di risorse MemoryDB specifici

La seguente politica consente esplicitamente l'accesso account-id completo specificato a tutte le risorse di tipo gruppo di sottorete, gruppo di sicurezza e cluster.

```
{
  "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/*"
  ]
}
```

Esempio 2: negare a un utente l'accesso a un cluster.

L'esempio seguente nega esplicitamente l'account -idaccesso specificato a un particolare cluster.

```
{
  "Sid": "Example2",
  "Effect": "Deny",
  "Action": "memorydb:*",
  "Resource": [
    "arn:aws:memorydb:us-east-1:account-id:cluster/name"
  ]
}
```

Utilizzo di ruoli collegati ai servizi per Amazon MemoryDB for Redis

[Amazon MemoryDB for Redis utilizza ruoli collegati ai servizi AWS Identity and Access Management \(IAM\)](#). Un ruolo collegato al servizio è un tipo unico di ruolo IAM collegato direttamente a un AWS servizio, come Amazon MemoryDB for Redis. I ruoli collegati ai servizi di Amazon MemoryDB for Redis sono predefiniti da Amazon MemoryDB for Redis. Includono tutte le autorizzazioni necessarie al servizio per richiamare servizi AWS per conto dei cluster.

Un ruolo collegato al servizio semplifica la configurazione di Amazon MemoryDB for Redis perché non è necessario aggiungere manualmente le autorizzazioni necessarie. I ruoli esistono già nel tuo AWS account ma sono collegati ai casi d'uso di Amazon MemoryDB for Redis e dispongono di autorizzazioni predefinite. Solo Amazon MemoryDB for Redis può assumere questi ruoli e solo questi ruoli possono utilizzare la politica di autorizzazioni predefinita. È possibile eliminare i ruoli solo dopo aver eliminato le risorse correlate. Questo protegge le tue risorse Amazon MemoryDB for Redis perché non puoi rimuovere inavvertitamente le autorizzazioni necessarie per accedere alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta la sezione [Servizi AWS che funzionano con IAM](#) e cerca i servizi che riportano Sì nella colonna Ruolo associato ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Indice

- [Autorizzazioni di ruolo collegate ai servizi per Amazon MemoryDB for Redis](#)
- [Creazione di un ruolo collegato ai servizi \(IAM\)](#)
 - [Creazione di un ruolo collegato ai servizi \(Console di IAM\)](#)
 - [Creazione di un ruolo collegato ai servizi \(CLI di IAM\)](#)

- [Creazione di un ruolo collegato ai servizi \(API di IAM\)](#)
- [Modifica della descrizione di un ruolo collegato ai servizi per Amazon MemoryDB for Redis](#)
 - [Modifica della descrizione di un ruolo collegato ai servizi \(console di IAM\)](#)
 - [Modifica della descrizione di un ruolo collegato ai servizi \(CLI di IAM\)](#)
 - [Modifica della descrizione di un ruolo collegato ai servizi \(API di IAM\)](#)
- [Eliminazione di un ruolo collegato ai servizi per Amazon MemoryDB for Redis](#)
 - [Pulizia di un ruolo collegato ai servizi](#)
 - [Eliminazione di un ruolo collegato ai servizi \(console di IAM\)](#)
 - [Eliminazione di un ruolo collegato ai servizi \(CLI di IAM\)](#)
 - [Eliminazione di un ruolo collegato ai servizi \(API di IAM\)](#)

Autorizzazioni di ruolo collegate ai servizi per Amazon MemoryDB for Redis

Amazon MemoryDB for Redis utilizza il ruolo collegato al servizio denominato `AWSServiceRoleForMemoryDB`: questa policy consente a MemoryDB di gestire AWS le risorse per tuo conto, se necessario per la gestione dei cluster.

La politica `AWSServiceRoleForMemoryDB` di autorizzazione dei ruoli collegati al servizio consente ad Amazon MemoryDB for Redis di completare le seguenti azioni sulle risorse specificate:

```
{
  "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"
        }
      }
    }
  ]
}

```

Per ulteriori informazioni, consulta [AWSpolitica gestita: MemoryDBServiceRolePolicy](#).

Per consentire a un'entità IAM di creare ruoli collegati ai servizi AWSServiceRoleForMemoryDB

Aggiungi la seguente istruzione di policy alle autorizzazioni per l'entità 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"}}
}

```

Per consentire a un'entità IAM di eliminare i ruoli collegati ai servizi AWSServiceRoleForMemoryDB

Aggiungi la seguente istruzione di policy alle autorizzazioni per l'entità 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"}}
}
```

In alternativa, puoi utilizzare una policy AWS gestita per fornire l'accesso completo ad Amazon MemoryDB for Redis.

Creazione di un ruolo collegato ai servizi (IAM)

È possibile creare un ruolo collegato ai servizi utilizzando la console di IAM, la CLI o l'API.

Creazione di un ruolo collegato ai servizi (Console di IAM)

Puoi utilizzare la console IAM per creare un ruolo collegato ai servizi.

Come creare un ruolo collegato ai servizi (console)

1. [Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
2. Nel riquadro di navigazione a sinistra della console IAM, scegli Ruoli. Quindi seleziona Create new role (Crea nuovo ruolo).
3. In Select type of trusted entity (Seleziona tipo di entità attendibile), scegli Service AWS .
4. In Oppure seleziona un servizio per visualizzarne i casi d'uso, scegli MemoryDB.
5. Scegli Successivo: Autorizzazioni.
6. In Policy name (Nome policy), si noti che MemoryDBServiceRolePolicy è necessario per questo ruolo. Scegli Successivo: Tag.
7. Si noti che i tag non sono supportati per i ruoli collegati al servizio. Scegliere Next:Review (Successivo:Rivedi).
8. (Facoltativo) In Role description (Descrizione ruolo) modifica la descrizione per il nuovo ruolo collegato ai servizi.
9. Rivedere il ruolo e scegliere Crea ruolo.

Creazione di un ruolo collegato ai servizi (CLI di IAM)

Puoi utilizzare le operazioni IAM di AWS Command Line Interface per creare un ruolo collegato al servizio. Questo ruolo può includere la policy di attendibilità e le policy inline che il servizio richiede per assumere il ruolo.

Per creare un ruolo collegato ai servizi (CLI)

Attenersi alle operazioni seguenti:

```
$ aws iam create-service-linked-role --aws-service-name memorydb.amazonaws.com
```

Creazione di un ruolo collegato ai servizi (API di IAM)

È possibile utilizzare l'API di IAM per creare un ruolo collegato ai servizi. Questo ruolo può contenere la policy di attendibilità e le policy inline che il servizio richiede per assumere il ruolo.

Per creare un ruolo collegato ai servizi (API)

Utilizzare la chiamata API [CreateServiceLinkedRole](#). Nella richiesta, specificare un nome del servizio di `memorydb.amazonaws.com`.

Modifica della descrizione di un ruolo collegato ai servizi per Amazon MemoryDB for Redis

Amazon MemoryDB for Redis non consente di modificare il `AWSServiceRoleForMemoryDB` ruolo collegato al servizio. Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM.

Modifica della descrizione di un ruolo collegato ai servizi (console di IAM)

È possibile utilizzare la console di IAM per modificare la descrizione di un ruolo collegato ai servizi.

Per modificare la descrizione di un ruolo collegato ai servizi (console)

1. Nel riquadro di navigazione a sinistra della console IAM, scegli Ruoli.
2. Scegliere il nome del ruolo da modificare.
3. Nella parte destra di Role description (Descrizione ruolo), scegliere Edit (Modifica).
4. Digita una nuova descrizione nella casella e scegli Save (Salva).

Modifica della descrizione di un ruolo collegato ai servizi (CLI di IAM)

Puoi utilizzare le operazioni IAM da AWS Command Line Interface per modificare una descrizione del ruolo collegato al servizio.

Per modificare la descrizione di un ruolo collegato ai servizi (CLI)

1. (Facoltativo) Per visualizzare la descrizione corrente di un ruolo, utilizza l'operazione AWS CLI for IAM. [get-role](#)

Example

```
$ aws iam get-role --role-name AWSServiceRoleForMemoryDB
```

Utilizzare il nome del ruolo, non l'ARN, per fare riferimento ai ruoli con le operazioni CLI. Ad esempio, per fare riferimento a un ruolo il cui ARN è `arn:aws:iam::123456789012:role/myrole`, puoi usare **myrole**.

2. Per aggiornare la descrizione di un ruolo collegato al servizio, utilizza l'operazione AWS CLI for IAM. [update-role-description](#)

Per Linux, macOS o Unix:

```
$ aws iam update-role-description \  
  --role-name AWSServiceRoleForMemoryDB \  
  --description "new description"
```

Per Windows:

```
$ aws iam update-role-description ^\  
  --role-name AWSServiceRoleForMemoryDB ^\  
  --description "new description"
```

Modifica della descrizione di un ruolo collegato ai servizi (API di IAM)

È possibile utilizzare l'API di IAM per modificare la descrizione di un ruolo collegato ai servizi.

Per modificare la descrizione di un ruolo collegato ai servizi (API)

1. (Facoltativo) Per visualizzare la descrizione corrente di un ruolo, utilizzare l'operazione API di IAM [GetRole](#).

Example

```
https://iam.amazonaws.com/
```

```
?Action=GetRole
&RoleName=AWSServiceRoleForMemoryDB
&Version=2010-05-08
&AUTHPARAMS
```

2. Per aggiornare la descrizione di un ruolo, utilizzare l'operazione API di IAM [UpdateRoleDescription](#).

Example

```
https://iam.amazonaws.com/
?Action=UpdateRoleDescription
&RoleName=AWSServiceRoleForMemoryDB
&Version=2010-05-08
&Description="New description"
```

Eliminazione di un ruolo collegato ai servizi per Amazon MemoryDB for Redis

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato ai servizi prima di poterlo eliminare.

Amazon MemoryDB for Redis non elimina automaticamente il ruolo collegato al servizio.

Pulizia di un ruolo collegato ai servizi

Prima di poter utilizzare IAM per eliminare un ruolo collegato al servizio, verifica innanzitutto che al ruolo non siano associate risorse (cluster).

Per verificare se il ruolo collegato ai servizi dispone di una sessione attiva nella console IAM

1. [Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
2. Nel riquadro di navigazione a sinistra della console IAM, scegli Ruoli. Quindi scegli il nome (non la casella di controllo) del *AWSServiceRoleForMemoryDB* ruolo.
3. Nella pagina Summary (Riepilogo) per il ruolo selezionato, scegliere la scheda Access Advisor (Consulente accessi).
4. Nella scheda Access Advisor (Consulente accessi), esamina l'attività recente per il ruolo collegato ai servizi.

Per eliminare le risorse di Amazon MemoryDB for Redis che richiedono AWSServiceRoleForMemoryDB (console)

- Per eliminare un cluster, consultare i seguenti argomenti:
 - [Usando il AWS Management Console](#)
 - [Usando il AWS CLI](#)
 - [Utilizzo dell'API MemoryDB](#)

Eliminazione di un ruolo collegato ai servizi (console di IAM)

È possibile utilizzare la console IAM per eliminare un ruolo collegato ai servizi.

Per eliminare un ruolo collegato ai servizi (console)

1. [Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.](https://console.aws.amazon.com/iam/)
2. Nel riquadro di navigazione a sinistra della console IAM, scegli Ruoli. Quindi, seleziona la casella di controllo accanto al nome del ruolo che desideri eliminare, non il nome o la riga stessa.
3. In operazioni Role (Ruolo) nella parte superiore della pagina, seleziona Delete (Elimina) ruolo.
4. Nella pagina di conferma, esamina i dati dell'ultimo accesso al servizio, che mostrano l'ultima volta che ciascuno dei ruoli selezionati ha effettuato l'ultimo accesso a un AWS servizio. In questo modo potrai verificare se il ruolo è attualmente attivo. Se desideri procedere, seleziona Yes, Delete (Sì, elimina) per richiedere l'eliminazione del ruolo collegato ai servizi.
5. Controlla le notifiche della console IAM per monitorare lo stato dell'eliminazione del ruolo collegato ai servizi. Poiché l'eliminazione del ruolo collegato ai servizi IAM è asincrona, una volta richiesta l'eliminazione del ruolo, il task di eliminazione può essere eseguito correttamente o meno. Se il task non viene eseguito correttamente, puoi scegliere View details (Visualizza dettagli) o View Resources (Visualizza risorse) dalle notifiche per capire perché l'eliminazione non è stata effettuata.

Eliminazione di un ruolo collegato ai servizi (CLI di IAM)

Puoi utilizzare le operazioni IAM da AWS Command Line Interface per eliminare un ruolo collegato al servizio.

Per eliminare un ruolo collegato ai servizi (CLI)

1. Se non conosci il nome del ruolo collegato ai servizi da eliminare, inserisci il comando seguente: Questo comando elenca i ruoli e i relativi Amazon Resource Name (ARN) nel tuo account.

```
$ aws iam get-role --role-name role-name
```

Utilizzare il nome del ruolo, non l'ARN, per fare riferimento ai ruoli con le operazioni CLI. Ad esempio, per fare riferimento a un ruolo il cui ARN è `arn:aws:iam::123456789012:role/myrole`, puoi usare **myrole**.

2. Poiché un ruolo collegato ai servizi non può essere eliminato se è in uso o se a esso sono associate delle risorse, occorre inviare una richiesta di eliminazione. Se queste condizioni non sono soddisfatte, la richiesta può essere rifiutata. Acquisisci il valore di `deletion-task-id` dalla risposta per controllare lo stato del task di eliminazione. Per inviare una richiesta di eliminazione per un ruolo collegato ai servizi, inserire quanto segue:

```
$ aws iam delete-service-linked-role --role-name role-name
```

3. Inserire quanto segue per verificare lo stato del processo di eliminazione:

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

Lo stato di un task di eliminazione può essere `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` o `FAILED`. Se l'eliminazione non viene eseguita correttamente, la chiamata restituisce il motivo dell'errore per consentire all'utente di risolvere il problema.

Eliminazione di un ruolo collegato ai servizi (API di IAM)

È possibile utilizzare l'API di IAM; per eliminare un ruolo collegato ai servizi.

Per eliminare un ruolo collegato ai servizi (API)

1. Per inviare una richiesta di eliminazione per un ruolo collegato ai servizi, chiamare [DeleteServiceLinkedRole](#). Nella richiesta, specifica il nome del ruolo.

Poiché un ruolo collegato ai servizi non può essere eliminato se è in uso o se a esso sono associate delle risorse, occorre inviare una richiesta di eliminazione. Se queste condizioni non

sono soddisfatte, la richiesta può essere rifiutata. Acquisisci il valore di `DeletionTaskId` dalla risposta per controllare lo stato del task di eliminazione.

2. Chiamare [GetServiceLinkedRoleDeletionStatus](#) per controllare lo stato dell'eliminazione. Nella richiesta, specificare il `DeletionTaskId`.

Lo stato di un task di eliminazione può essere `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` o `FAILED`. Se l'eliminazione non viene eseguita correttamente, la chiamata restituisce il motivo dell'errore per consentire all'utente di risolvere il problema.

AWS policy gestite per MemoryDB for Redis

Per aggiungere le autorizzazioni a utenti, gruppi e ruoli, è più semplice utilizzare policy gestite da AWS piuttosto che scrivere autonomamente le policy. La [creazione di policy gestite dai clienti IAM](#) che forniscono al tuo team solo le autorizzazioni di cui ha bisogno richiede tempo e competenza. Per iniziare rapidamente, utilizza le nostre policy gestite da AWS. Queste policy coprono i casi d'uso più comuni e sono disponibili nel tuo account AWS. Per ulteriori informazioni sulle policy gestite da AWS, consulta [Policy gestite da AWS](#) nella Guida per l'utente di IAM.

I servizi AWS mantengono e aggiornano le policy gestite da AWS. Non è possibile modificare le autorizzazioni nelle policy gestite da AWS. I servizi occasionalmente aggiungono altre autorizzazioni a una policy gestita da AWS per supportare nuove funzionalità. Questo tipo di aggiornamento interessa tutte le identità (utenti, gruppi e ruoli) a cui è collegata la policy. È più probabile che i servizi aggiornino una policy gestita da AWS quando viene avviata una nuova funzionalità o quando diventano disponibili nuove operazioni. I servizi non rimuovono le autorizzazioni da una policy gestita da AWS, pertanto gli aggiornamenti delle policy non interrompono le autorizzazioni esistenti.

Inoltre, AWS supporta policy gestite per le funzioni di processi che coprono più servizi. Ad esempio, la policy `ReadOnlyAccessAWS` gestita da fornisce accesso in sola lettura a tutti i servizi AWS e le risorse. Quando un servizio avvia una nuova funzionalità, AWS aggiunge autorizzazioni di sola lettura per nuove operazioni e risorse. Per l'elenco e la descrizione delle policy di funzione dei processi, consulta la sezione [Policy gestite da AWS per funzioni di processi](#) nella Guida per l'utente di IAM.

AWS politica gestita: `MemoryDBServiceRolePolicy`

Non puoi allegare la politica `ServiceRolePolicy` AWS gestita da MemoryDB alle identità del tuo account. Questa policy fa parte del ruolo collegato ai servizi AWS MemoryDB. Questo ruolo consente al servizio di gestire le interfacce di rete e i gruppi di sicurezza nel tuo account.

MemoryDB utilizza le autorizzazioni di questa politica per gestire i gruppi di sicurezza EC2 e le interfacce di rete. Ciò è necessario per gestire i cluster MemoryDB.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

```
{
  "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"
        }
      }
    }
  }

```

```
]
}
```

AWSPolicy gestite da (predefinite) per MemoryDB for Redis

AWS gestisce molti casi di utilizzo comune fornendo policy IAM autonome create e amministrare da AWS. Le policy gestite concedono le autorizzazioni necessarie per i casi di utilizzo comune in modo da non dover cercare quali sono le autorizzazioni richieste. Per ulteriori informazioni, consulta [Policy gestite da AWS](#) nella Guida per l'utente di IAM.

Le seguenti policyAWS gestite da, che puoi collegare agli utenti nel tuo account, sono specifiche di MemoryDB:

AmazonMemoryDBReadOnlyAccess

È possibile allegare la policy `AmazonMemoryDBReadOnlyAccess` alle identità IAM. Questa policy concede le autorizzazioni amministrative che consentono l'accesso in sola lettura a tutte le risorse MemoryDB.

`AmazonMemoryDBReadOnlyAccess`: concede l'accesso in sola lettura alle risorse di MemoryDB for Redis.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "memorydb:Describe*",
      "memorydb:List*"
    ],
    "Resource": "*"
  }]
}
```

AmazonMemoryDBFullAccess

È possibile allegare la policy `AmazonMemoryDBFullAccess` alle identità IAM. Questa policy concede le autorizzazioni amministrative che consentono l'accesso completo a tutte le risorse MemoryDB.

`AmazonMemoryDBFullAccess`: garantisce l'accesso completo alle risorse di MemoryDB for 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"
      }
    }
  }
]
}
```

Puoi anche creare policy IAM personalizzate per concedere autorizzazioni per operazioni API MemoryDB for Redis. Puoi collegare queste policy personalizzate agli utenti o ai gruppi IAM che richiedono le autorizzazioni.

Aggiornamenti di MemoryDB alle policyAWS gestite

Visualizza i dettagli sugli aggiornamenti alle policyAWS gestite da per MemoryDB da quando questo servizio ha iniziato a tenere traccia delle modifiche. Per gli avvisi automatici sulle modifiche apportate a questa pagina, sottoscrivi il feed RSS nella pagina della cronologia dei documenti di MemoryDB.

Modifica	Descrizione	Data
AmazonMemoryDBFull Access — Aggiungere una politica	MemoryDB ha aggiunto nuove autorizzazioni per descrivere ed elencare le risorse supportate. Queste autorizzazioni sono necessarie per	10/07/2021

Modifica	Descrizione	Data
	MemoryDB per eseguire query su tutte le risorse supportate in un account.	
AmazonMemoryDBReadOnlyAccess — Aggiungere una politica	MemoryDB ha aggiunto nuove autorizzazioni per descrivere ed elencare le risorse supportate. Queste autorizzazioni sono necessari e per MemoryDB per creare applicazioni basate su account eseguendo query su tutte le risorse supportate in un account.	10/07/2021
MemoryDB ha iniziato a monitorare le modifiche	Avvio del servizio	19/08/2021

Autorizzazioni API MemoryDB: riferimento ad azioni, risorse e condizioni

Quando configuri le policy di [controllo degli accessi](#) e di scrittura per le autorizzazioni da allegare a una policy IAM (basata sull'identità o basata sulle risorse), utilizza la tabella seguente come riferimento. La tabella elenca ogni operazione dell'API MemoryDB for Redis e le azioni corrispondenti per le quali è possibile concedere le autorizzazioni per eseguire l'azione. Puoi specificare le operazioni nel campo `Action` della policy e il valore di una risorsa nel campo `Resource` della policy. Se non diversamente indicato, la risorsa è obbligatoria. Alcuni campi includono sia una risorsa obbligatoria che risorse facoltative. Quando non è presente alcuna risorsa ARN, la risorsa nella policy è un carattere jolly (*).

Note

Per specificare un'operazione, utilizza il prefisso `memorydb:` seguito dal nome dell'operazione API (ad esempio, `memorydb:DescribeClusters`).

Registrazione di log e monitoraggio

Il monitoraggio è una parte importante del mantenimento dell'affidabilità, della disponibilità e delle prestazioni di MemoryDB for Redis e delle altre soluzioni. AWS fornisce i seguenti strumenti di monitoraggio per monitorare MemoryDB, segnalare quando qualcosa non va e intraprendere azioni automatiche quando appropriato:

- Amazon CloudWatch monitora AWS le tue risorse e le applicazioni su cui esegui AWS in tempo reale. Puoi raccogliere i parametri e tenerne traccia, creare pannelli di controllo personalizzati e impostare allarmi per inviare una notifica o intraprendere azioni quando un parametro specificato raggiunge una determinata soglia. Ad esempio, puoi tenere CloudWatch traccia dell'utilizzo della CPU o di altri parametri delle tue istanze Amazon EC2 e avviare automaticamente nuove istanze quando necessario. Per ulteriori informazioni, consulta la [Amazon CloudWatch User Guide](#).
- Amazon CloudWatch Logs ti consente di monitorare, archiviare e accedere ai tuoi file di log da istanze Amazon EC2 e altre CloudTrail fonti. CloudWatch Logs possono monitorare le informazioni nei file di registro e avvisarti quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la [Amazon CloudWatch Logs User Guide](#).
- AWS CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo AWS account e invia i file di log a un bucket Amazon S3 da te specificato. Puoi identificare quali

utenti e account hanno chiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute le chiamate. Per ulteriori informazioni, consulta la [Guida per l'utente AWS CloudTrail](#).

Monitoraggio di MemoryDB per Redis con Amazon CloudWatch

Puoi monitorare MemoryDB for Redis utilizzando CloudWatch, che raccoglie dati grezzi e li elabora in metriche leggibili e quasi in tempo reale. Queste statistiche vengono conservate per un periodo di 15 mesi, per permettere l'accesso alle informazioni storiche e offrire una prospettiva migliore sulle prestazioni del servizio o dell'applicazione web. È anche possibile impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando queste soglie vengono raggiunte. Per ulteriori informazioni, consulta la [Amazon CloudWatch User Guide](#).

Le seguenti sezioni elencano le metriche e le dimensioni per MemoryDB.

Argomenti

- [Parametri a livello di host](#)
- [Metriche per MemoryDB](#)
- [Quali parametri è opportuno monitorare?](#)
- [Scelta delle statistiche e dei periodi di un parametro](#)
- [Metriche di monitoraggio CloudWatch](#)

Parametri a livello di host

Il AWS/MemoryDB namespace include le seguenti metriche a livello di host per i singoli nodi.

Vedi anche

- [Metriche per MemoryDB](#)

Parametro	Descrizione	Unità
CPUUtilization	La percentuale di utilizzo CPU per l'intero host. Poiché Redis è a thread singolo, ti consigliamo di monitorare la EngineCPUUtilization metrica per i nodi con 4 o più vCPU.	Percentuale

Parametro	Descrizione	Unità
FreeableMemory	La quantità di memoria libera disponibile sull'host. Questo è derivato dalla RAM e dai buffer e dal fatto che il sistema operativo riporta come liberabile.	Byte
NetworkBytesIn	Il numero di byte che l'host ha letto dalla rete.	Byte
NetworkBytesOut	Il numero di byte inviati dall'istanza su tutte le interfacce di rete.	Byte
NetworkPacketsIn	Il numero di pacchetti ricevuti dall'istanza su tutte le interfacce di rete. Questo parametro identifica il volume del traffico in entrata in termini di numero di pacchetti su una singola istanza.	Conteggio
NetworkPacketsOut	Il numero di pacchetti inviati dall'istanza su tutte le interfacce di rete. Questo parametro identifica il volume del traffico in uscita in termini di numero di pacchetti su una singola istanza.	Conteggio
NetworkBandwidthInAllowanceExceeded	Numero di pacchetti modellati perché la larghezza di banda aggregata in entrata ha superato il valore massimo per l'istanza.	Conteggio
NetworkConntrackAllowanceExceeded	Numero di pacchetti modellati perché il rilevamento delle connessioni ha superato il valore massimo per l'istanza e non è stato possibile stabilire nuove connessioni. Ciò può comportare la perdita di pacchetti per il traffico da o verso l'istanza.	Conteggio
NetworkBandwidthOutAllowanceExceeded	Numero di pacchetti modellati perché la larghezza di banda aggregata in uscita ha superato il valore massimo per l'istanza.	Conteggio

Parametro	Descrizione	Unità
NetworkPacketsPerSecondAllowanceExceeded	Numero di pacchetti modellati perché i pacchetti bidirezionali al secondo hanno superato il valore massimo per l'istanza.	Conteggio
NetworkMaxBytesIn	Il burst massimo di byte ricevuti ogni minuto.	Byte
NetworkMaxBytesOut	Il burst massimo di byte trasmessi ogni minuto.	Byte
NetworkMaxPacketsIn	Il burst massimo di pacchetti ricevuti ogni minuto.	Conteggio
NetworkMaxPacketsOut	Il burst massimo di pacchetti trasmessi ogni minuto.	Conteggio
SwapUsage	La quantità di spazio di swapping utilizzato sull'host.	Byte

Metriche per MemoryDB

Il namespace AWS/MemoryDB include i parametri di Redis descritti di seguito.

Ad eccezione di `ReplicationLag` e `EngineCPUUtilization`, questi parametri sono derivati dal comando `info` di Redis. Ogni metrica viene calcolata a livello di nodo.

Per una documentazione completa del comando `info` di Redis, visita la pagina all'indirizzo <http://redis.io/commands/info>.

Vedi anche

- [Parametri a livello di host](#)


Parametro	Descrizione	Unità
ActiveDefragHits	Il numero di riallocazioni di valori al minuto effettuate dal processo di deframmentazione	Numero

Parametro	Descrizione	Unità
	attivo. È derivato dalla statistica <code>active_defrag_hits</code> in Redis INFO .	
AuthenticationFailures	Numero totale di tentativi non riusciti di autenticazione su Redis utilizzando il comando AUTH. Ulteriori informazioni sui singoli errori di autenticazione sono disponibili tramite il comando LOG DI CONTROLLO ACCESSI . Consigliamo di impostare un allarme per rilevare i tentativi di accesso non autorizzato.	Conteggio
	Il numero totale di byte allocati da MemoryDB per tutti gli scopi, inclusi il set di dati, i buffer e così via.	Byte
BytesUsedForMemoryDB	Dimension: Tier=SSD per i cluster che utilizzano Tiering di dati : il numero totale di byte utilizzati dall'SSD.	Byte
	Dimension: Tier=Memory per i cluster che utilizzano Tiering di dati : il numero totale di byte utilizzati dalla memoria. Questo è il valore di statistica <code>used_memory</code> in Informazioni su Redis .	Byte
BytesReadFromDisk	Il numero totale di byte letti dal disco al minuto. Supportata solo per i cluster che utilizzano Tiering di dati .	Byte
BytesWrittenToDisk	Il numero totale di byte scritti su disco al minuto. Supportata solo per i cluster che utilizzano Tiering di dati .	Byte

Parametro	Descrizione	Unità
<code>CommandAuthorizationFailures</code>	Numero totale di tentativi non riusciti da parte degli utenti di eseguire comandi che non dispongono dell'autorizzazione per chiamare. Ulteriori informazioni sui singoli errori di autenticazione sono disponibili tramite il comando LOG DI CONTROLLO ACCESSI . Consigliamo di impostare un allarme per rilevare i tentativi di accesso non autorizzato.	Conteggio
<code>CurrConnections</code>	Numero di connessioni client, escluse le connessioni dalle repliche di lettura. MemoryDB utilizza da due a quattro connessioni per monitorare il cluster in ogni caso. È derivato dalla statistica <code>connected_clients</code> in Redis INFO .	Conteggio
<code>CurrItems</code>	Il numero di elementi nella cache. È derivato dalla statistica <code>keyspace</code> di Redis sommando tutte le chiavi nell'intero keyspace.	Conteggio
	Dimension: <code>Tier=Memory</code> per cluster che utilizzano Tiering di dati . Il numero di elementi in memoria.	Conteggio
<code>DatabaseMemoryUsagePercentage</code>	Dimension: <code>Tier=SSD (Solid State Drive)</code> per cluster che utilizzano Tiering di dati . Il numero di elementi nell'SSD.	Conteggio
	Percentuale della memoria disponibile per il cluster in uso. Questo viene calcolato utilizzando <code>used_memory/maxmemory</code> da Redis INFO .	Percentuale

Parametro	Descrizione	Unità
DatabaseCapacityUsagePercentage	<p>Percentuale della capacità di dati totale disponibile per il cluster in uso.</p> <p><u>Nelle istanze Data Tiered, la metrica viene calcolata come $(\text{used_memory} - \text{mem_not_counted_for_evict} + \text{SSD used}) / (\text{maxmemory} + \text{SSD total capacity})$, dove <code>used_memory</code> e <code>maxmemory</code> viene presa da Redis INFO.</u></p> <p>In tutti gli altri casi, la metrica viene calcolata utilizzando <code>used_memory/maxmemory</code></p>	Percentuale
DB0AverageTTL	<p>Esponde <code>avg_ttl</code> di DBO dalla statistica <code>keyspace</code> del comando INFO Redis.</p>	Millisecondi

Parametro	Descrizione	Unità
EngineCPUUtilization	<p>Fornisce l'utilizzo della CPU del thread del motore Redis. Poiché Redis è a thread singolo, è possibile utilizzare questo parametro per analizzare il carico del processo Redis stesso. Il parametro EngineCPUUtilization fornisce una visibilità più precisa del processo Redis. Lo si può usare assieme al parametro CPUUtilization . CPUUtilization espone l'utilizzo della CPU per l'istanza del server nel complesso, inclusi altri processi di gestione e legati ai sistemi operativi. Per tipi di nodi più grandi con almeno quattro vCPU, utilizza il parametro EngineCPUUtilization per monitorare e impostare le soglie per il dimensionamento.</p>	Percentuale

 **Note**

Su un host MemoryDB, i processi in background monitorano l'host per fornire un'esperienza di database gestita. Questi processi in background possono occupare una parte significativa del carico di lavoro della CPU. Questo non è significativo per gli host più grandi con più di due vCPU. Ma può interessare host più piccoli con 2 vCPU o meno. Se si monitora solo la metrica EngineCPUUtilization , non si sarà a conoscenza di situazioni in cui l'host è sovraccaricato sia con un utilizzo elevato della CPU da Redis che con un utilizzo elevato della CPU dai processi di monitoraggio in background. Pertanto, si consiglia di monitorare il

Parametro	Descrizione	Unità
	parametro <code>CPUtilization</code> per gli host con due vCPU o meno.	
<code>Evictions</code>	Il numero di chiavi che sono state rimosse a causa del limite <code>maxmemory</code> . È derivato dalla statistica <code>evicted_keys</code> in Redis INFO .	Conteggio
<code>IsPrimary</code>	Indica se il nodo è il nodo primario dello shard corrente. Il parametro può essere 0 (non primario) o 1 (primario).	Conteggio
<code>KeyAuthorizationFailures</code>	Numero totale di tentativi non riusciti da parte degli utenti di accedere a chiavi a cui non hanno l'autorizzazione ad accedere. Ulteriori informazioni sui singoli errori di autenticazione sono disponibili tramite il comando LOG DI CONTROLLO ACCESSI . Consigliamo di impostare un allarme per rilevare i tentativi di accesso non autorizzato.	Conteggio
<code>KeyspaceHits</code>	Il numero di ricerche di chiavi di sola lettura nella directory principale. È derivato dalla statistica <code>keyspace_hits</code> in Redis INFO .	Conteggio
<code>KeyspaceMisses</code>	Il numero di ricerche di chiavi di sola lettura non riuscite nella directory principale. È derivato dalla statistica <code>keyspace_misses</code> in Redis INFO .	Conteggio

Parametro	Descrizione	Unità
KeysTracked	Il numero di chiavi tracciate dal rilevamento delle chiavi Redis come percentuale di <code>tracking-table-max-keys</code> . Il tracciamento delle chiavi viene utilizzato per facilitare e la memorizzazione nella cache lato client e notifica ai client quando le chiavi vengono modificate.	Conteggio
MaxReplicationThroughput	La velocità di replica massima osservata durante l'ultimo ciclo di misurazione.	Byte al secondo
MemoryFragmentationRatio	Indica l'efficienza nell'allocazione della memoria del motore Redis. Dalla soglia dipendono comportamenti diversi. Il valore consigliato è avere una frammentazione superiore a 1.0. Questo viene calcolato da <code>mem_fragmentation_ratio</code> statistic di Redis INFO .	Numero
NewConnections	Il numero totale di connessioni accettate dal server durante questo periodo. È derivato dalla statistica <code>total_connections_received</code> in Redis INFO .	Conteggio
NumItemsReadFromDisk	Il numero totale di elementi recuperati dal disco al minuto. Supportata solo per i cluster che utilizzano Tiering di dati .	Conteggio
NumItemsWrittenToDisk	Il numero totale di elementi scritti su disco al minuto. Supportata solo per i cluster che utilizzano Tiering di dati .	Conteggio

Parametro	Descrizione	Unità
PrimaryLinkHealthStatus	Questo stato ha due valori: 0 e 1. Il valore 0 indica che i dati nel nodo primario di MemoryDB non sono sincronizzati con Redis su EC2. Il valore 1 indica che i dati sono in fase di sincronizzazione.	Booleano
Reclaimed	Il numero totale di eventi di scadenza di chiavi. È derivato dalla statistica <code>expired_keys</code> in Redis INFO .	Conteggio
ReplicationBytes	Per i nodi in una configurazione replicata, <code>ReplicationBytes</code> indica il numero di byte che il nodo primario sta inviando a tutte le relative repliche. Questa metrica è rappresentativa del carico di scrittura sul cluster. È derivato dalla statistica <code>master_repl_offset</code> in Redis INFO .	Byte
ReplicationDelayedWriteCommands	Numero di comandi di scrittura ritardati a causa della replica sincrona. La replica può essere ritardata a causa di vari fattori, ad esempio la congestione della rete o il superamento del throughput di replica massimo.	Conteggio
ReplicationLag	Questo parametro è applicabile soltanto per un nodo in esecuzione come replica di lettura. Rappresenta il ritardo, in secondi, della replica nell'applicare le modifiche generate dal nodo primario.	Secondi

Di seguito sono riportate le aggregazioni di certi tipi di comandi, derivati da `info commandstats`: La sezione `commandstats` fornisce statistiche basate sul tipo di comando, incluso il numero di chiamate.

Per l'elenco completo dei comandi disponibili, consulta [comandi Redis](#) nella documentazione relativa a Redis.

Parametro	Descrizione	Unità
EvalBasedCmds	Numero totale di comandi per i comandi basati su valutazione. Questo è derivato dalla statistica <code>commandstats</code> Redis. Questo è derivato dalla <code>commandstats</code> statistica sommando <code>eval</code> , <code>evalsha</code> di Redis.	Conteggio
GeoSpatialBasedCmds	Numero totale di comandi per i comandi basati su GeoSpace. Questo è derivato dalla statistica <code>commandstats</code> Redis. Viene ricavato sommando tutti i tipi di comandi geo: <code>geoadd</code> , <code>geodist</code> , <code>geohash</code> , <code>geopos</code> , <code>georadius</code> , e <code>georadiusbymember</code> .	Conteggio
GetTypeCmds	Il numero totale di comandi di tipo read-only. È ricavato dalla <code>commandstats</code> statistica sommando tutti i read-only comandi di tipo (<code>get</code> , <code>hget</code> , <code>scard</code> , <code>lrange</code> ecc.) di Redis.	Conteggio
HashBasedCmds	Il numero totale di comandi basati su hash. È derivato dalla statistica <code>commandstats</code> di Redis sommando tutti i comandi che agiscono su uno o più hash (<code>hget</code> , <code>hkeys</code> , <code>hvals</code> , <code>hdel</code> , etc).	Conteggio
HyperLogLogBasedCmds	Il numero totale di comandi basati su HyperLogLog. È ricavato dalla <code>commandstats</code> statistica sommando tutti i pf comandi di tipo (<code>pfadd</code> , <code>pfcount</code> , <code>pfmerge</code> , ecc.) di Redis.	Conteggio
JsonBasedCmds	Il numero totale di comandi basati su JSON. È ricavato dalla statistica <code>commandstats</code> di Redis sommando tutti i comandi che agiscono in base a uno o più oggetti documento JSON.	Conteggio
KeyBasedCmds	Il numero totale di comandi basati su chiavi. È derivato dalla statistica <code>commandstats</code> di	Conteggio

Parametro	Descrizione	Unità
	Redis sommando tutti i comandi che agiscono su una o più chiavi nelle strutture di dati multiple (del, expire, rename ecc.).	
ListBasedCmds	Il numero totale di comandi basati su elenchi. È derivato dalla statistica <code>commandstats</code> di Redis sommando tutti i comandi che agiscono su uno o più elenchi (lindex, lrange, lpush, ltrim, etc).	Conteggio
PubSubBasedCmds	Numero totale di comandi per la funzionalità pub/sub. Questo è derivato dalle <code>commandstats</code> statistiche di Redis sommando tutti i comandi utilizzati per la funzionalità pub/sub: <code>psubscribe</code> , <code>publish</code> , <code>pubsubpunsubscribe</code> , <code>subscribe</code> , <code>unsubscribe</code>	Conteggio
SearchBasedCmds	Il numero totale di comandi di indicizzazione e ricerca secondari, inclusi i comandi di lettura e scrittura. Questo è derivato dalla <code>commandstats</code> statistica di Redis sommando tutti i comandi di ricerca che agiscono sugli indici secondari.	Conteggio
SearchBasedGetCmds	Numero totale di comandi secondari di sola lettura per l'indice e la ricerca. Questo è derivato dalla <code>commandstats</code> statistica di Redis sommando tutti i comandi secondari di <code>index</code> e <code>search get</code> .	Conteggio
SearchBasedSetCmds	Numero totale di comandi secondari di indicizzazione e scrittura di ricerca. Questo è derivato dalla <code>commandstats</code> statistica di Redis sommando tutti i comandi secondari dell'indice e del set di ricerca.	Conteggio

Parametro	Descrizione	Unità
<code>SearchNumberOfIndices</code>	Numero totale di indici.	Conteggio
<code>SearchNumberOfIndexedKeys</code>	Numero totale di chiavi Redis indicizzate	Conteggio
<code>SearchTotalIndexSize</code>	Memoria (byte) utilizzata da tutti gli indici.	Byte
<code>SetBasedCmds</code>	Il numero totale di comandi basati su set. È derivato dalla statistica <code>commandstats</code> di Redis sommando tutti i comandi che agiscono su uno o più set (<code>scard</code> , <code>sdiff</code> , <code>sadd</code> , <code>sunion</code> , ecc.).	Conteggio
<code>SetTypeCmds</code>	Il numero totale di comandi di tipo write. È derivato dalla statistica <code>commandstats</code> di Redis sommando tutti i comandi di tipo mutative che operano sui dati (<code>set</code> , <code>hset</code> , <code>sadd</code> , <code>lpop</code> ecc.).	Conteggio
<code>SortedSetBasedCmds</code>	Il numero totale di comandi basati su set ordinati. È derivato dalla statistica <code>commandstats</code> di Redis sommando tutti i comandi che agiscono su uno o più set ordinati (<code>zcount</code> , <code>zrange</code> , <code>zrank</code> , <code>zadd</code> , ecc.).	Conteggio
<code>StringBasedCmds</code>	Il numero totale di comandi basati su stringhe. È derivato dalla statistica <code>commandstats</code> di Redis sommando tutti i comandi che agiscono su una o più stringhe (<code>strlen</code> , <code>setex</code> , <code>setrange</code> , ecc.).	Conteggio

Parametro	Descrizione	Unità
StreamBasedCmds	Il numero totale di comandi basati sul flusso. È derivato dalla statistica <code>commandstats</code> di Redis sommando tutti i comandi che agiscono su uno o più tipi di dati di flusso (<code>xrange</code> , <code>xlen</code> , <code>xadd</code> , <code>xdel</code> , ecc.).	Conteggio

Quali parametri è opportuno monitorare?

Le seguenti CloudWatch metriche offrono informazioni approfondite sulle prestazioni di MemoryDB. Nella maggior parte dei casi, si consiglia di impostare CloudWatch allarmi per queste metriche in modo da poter intraprendere azioni correttive prima che si verifichino problemi di prestazioni.

Parametri da monitorare

- [CPUUtilization](#)
- [EngineCPUUtilization](#)
- [SwapUsage](#)
- [Espulsioni](#)
- [CurrConnections](#)
- [Memoria](#)
- [Rete](#)
- [Replica](#)

CPUUtilization

Si tratta di un parametro a livello di host restituito sotto forma di percentuale. Per ulteriori informazioni, consulta [Parametri a livello di host](#).

Per i tipi di nodo più piccoli con al massimo 2 vCPU, utilizza il parametro `CPUUtilization` per monitorare il carico di lavoro.

In linea generale, ti consigliamo di impostare la soglia al 90% della CPU disponibile. Poiché Redis è a thread singolo, il valore di soglia effettivo deve essere calcolato come una frazione della capacità totale del nodo. Ad esempio, supponi che il tipo di nodo in uso supporti due core. In questo caso, la soglia per `CPUUtilization` sarebbe $90/2$ o 45%. [Per trovare il numero di core \(vCPU\) del tuo tipo di nodo, consulta i prezzi di MemoryDB](#).

Dovrai determinare la tua soglia, in base al numero di core nel nodo che stai utilizzando. Se superi questa soglia e il tuo carico di lavoro principale deriva dalle richieste di lettura, ridimensiona il cluster aggiungendo repliche di lettura. Se il carico di lavoro principale proviene da richieste di scrittura, ti consigliamo di aggiungere altri shard per distribuire il carico di lavoro di scrittura su più nodi primari.

Tip

Invece di utilizzare la metrica `Host-LevelCPUUtilization`, potresti essere in grado di utilizzare la metrica `RedisEngineCPUUtilization`, che riporta la percentuale di utilizzo sul core del motore Redis. [Per vedere se questa metrica è disponibile sui tuoi nodi e per ulteriori informazioni, consulta `Metrics for MemoryDB`.](#)

Per tipi di nodi più grandi con almeno 4 vCPU, puoi utilizzare il parametro `EngineCPUUtilization` che indica la percentuale di utilizzo sul core del motore Redis. [Per vedere se questa metrica è disponibile sui tuoi nodi e per ulteriori informazioni, consulta `Metrics for MemoryDB`.](#)

EngineCPUUtilization

Per tipi di nodi più grandi con almeno 4 vCPU, puoi utilizzare il parametro `EngineCPUUtilization` che indica la percentuale di utilizzo sul core del motore Redis. [Per vedere se questa metrica è disponibile sui tuoi nodi e per ulteriori informazioni, consulta `Metrics for MemoryDB`.](#)

SwapUsage

Si tratta di un parametro a livello di host restituito in byte. Per ulteriori informazioni, consulta [Parametri a livello di host](#).

Questo parametro non deve superare 50 MB.

Espulsioni

Questa è una metrica del motore. Ti consigliamo di determinare la tua soglia di allarme per questo parametro in base alle esigenze dell'applicazione.

CurrConnections

Questa è una metrica del motore. Ti consigliamo di determinare la tua soglia di allarme per questo parametro in base alle esigenze dell'applicazione.

Un numero crescente di `CurrConnections` dati potrebbe indicare un problema con l'applicazione; per risolvere il problema, sarà necessario esaminare il comportamento dell'applicazione.

Memoria

La memoria è un aspetto fondamentale di Redis. È necessario comprendere l'utilizzo della memoria del cluster per evitare la perdita di dati e consentire la crescita futura del set di dati. Le statistiche

sull'utilizzo della memoria di un nodo sono disponibili nella sezione di memoria del comando di Redis [INFO](#).

Rete

Uno dei fattori determinanti per la capacità della larghezza di banda di rete del cluster è il tipo di nodo selezionato. Per ulteriori informazioni sulla capacità di rete del tuo nodo, consulta i prezzi di [Amazon MemoryDB](#).

Replica

Il volume dei dati da replicare è visibile tramite il parametro `ReplicationBytes`. Puoi monitorare la velocità effettiva `MaxReplicationThroughput` della capacità di replica. Si consiglia di aggiungere altri shard quando si raggiunge il throughput massimo della capacità di replica.

`ReplicationDelayedWriteCommands` può anche indicare se il carico di lavoro supera il throughput massimo della capacità di replica. [Per ulteriori informazioni sulla replica in MemoryDB, vedere Understanding MemoryDB replication](#)

Scelta delle statistiche e dei periodi di un parametro

Sebbene CloudWatch ti consenta di scegliere qualsiasi statistica e periodo per ogni metrica, non tutte le combinazioni saranno utili. Ad esempio, le statistiche media, minimo e massimo di CPUUtilization sono utili, ma la statistica somma non lo è.

Tutti gli esempi di MemoryDB vengono pubblicati per una durata di 60 secondi per ogni singolo nodo. Per ogni periodo di 60 secondi, una metrica del nodo conterrà solo un singolo campione.

Metriche di monitoraggio CloudWatch

MemoryDB e CloudWatch sono integrati in modo da poter raccogliere una varietà di metriche. Puoi monitorare queste metriche utilizzando CloudWatch

Note

Gli esempi seguenti richiedono gli strumenti della CloudWatch riga di comando. Per ulteriori informazioni CloudWatch e per scaricare gli strumenti per sviluppatori, consulta la [pagina CloudWatch del prodotto](#).

Le seguenti procedure mostrano come CloudWatch raccogliere le statistiche sullo spazio di archiviazione per un cluster nell'ultima ora.

Note

I EndTime valori StartTime and forniti negli esempi seguenti sono a scopo illustrativo. Assicurati di sostituire i valori dell'ora di inizio e di fine appropriati per i tuoi nodi.

Per informazioni sui limiti di MemoryDB, consulta i limiti del [AWS servizio](#) per MemoryDB.

Metriche di monitoraggio CloudWatch (Console)

Per raccogliere statistiche sull'utilizzo della CPU per un cluster

1. [Accedere AWS Management Console e aprire la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. Seleziona i nodi per i quali desideri visualizzare le metriche.

 Note

La selezione di oltre 20 nodi disabilita la visualizzazione dei parametri sulla console.

- a. Nella pagina Cluster della Console di AWS gestione, fai clic sul nome di uno o più cluster.
Viene visualizzata la pagina di dettaglio del cluster.
- b. Fare clic sulla scheda Nodes (Nodi) nella parte superiore della finestra.
- c. Nella scheda Nodi della finestra di dettaglio, seleziona i nodi per i quali desideri visualizzare le metriche.

Nella parte inferiore della finestra della console viene visualizzato un elenco di CloudWatch metriche disponibili.

- d. Fare clic sul parametro CPU Utilization (Utilizzo CPU).

La CloudWatch console si aprirà e mostrerà le metriche selezionate. È possibile modificare i parametri visualizzati, mediante gli elenchi a discesa di Statistic (Statistica) e Period (Periodo) e la scheda Time Range (Intervallo di tempo).

Monitoraggio delle CloudWatch metriche tramite la CLI CloudWatch

Per raccogliere statistiche sull'utilizzo della CPU per un cluster

- Utilizzate il CloudWatch comando `aws cloudwatch get-metric-statistics` con i seguenti parametri (notate che gli orari di inizio e fine sono mostrati solo a titolo di esempio; sarà necessario sostituirli con gli orari di inizio e di fine appropriati):

Per 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
```

Per 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
```

Monitoraggio delle CloudWatch metriche tramite l'API CloudWatch

Per raccogliere statistiche sull'utilizzo della CPU per un cluster

- Chiamate l' CloudWatch API `GetMetricStatistics` con i seguenti parametri (tenete presente che gli orari di inizio e fine sono mostrati solo a titolo di esempio; sarà necessario sostituire gli orari di inizio e fine appropriati):
 - `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>
```

Monitoraggio degli eventi di MemoryDB per Redis

Quando si verificano eventi significativi per un cluster, MemoryDB invia una notifica a un argomento specifico di Amazon SNS. Esempi includono l'impossibilità di aggiungere un nodo, l'aggiunta di un nodo, la modifica di un gruppo di sicurezza e altro ancora. Tramite il monitoraggio degli eventi chiave, è possibile conoscere lo stato corrente dei cluster e, in base all'evento, intraprendere eventuali operazioni correttive.

Argomenti

- [Gestione delle notifiche Amazon SNS di MemoryDB](#)
- [Visualizzazione degli eventi di MemoryDB](#)
- [Notifiche di eventi Amazon SNS](#)

Gestione delle notifiche Amazon SNS di MemoryDB

Puoi configurare MemoryDB per inviare notifiche per importanti eventi del cluster utilizzando Amazon Simple Notification Service (Amazon SNS). Negli esempi che seguono, verrà configurato un cluster con l'Amazon Resource Name (ARN) di un argomento Amazon SNS per la ricezione di notifiche.

Note

Tale argomento presuppone l'avvenuta registrazione a Amazon SNS, nonché la configurazione e sottoscrizione di un argomento Amazon SNS. Per ulteriori informazioni su come procedere, consultare la [Guida per gli sviluppatori di Amazon Simple Notification Service](#).

Aggiungere un argomento Amazon SNS.

Le seguenti sezioni mostrano come aggiungere un argomento Amazon SNS utilizzando la AWS console, l'API MemoryDB, AWS CLI

Aggiunta di un argomento Amazon SNS (console)

Nella seguente procedura viene mostrato come aggiungere un argomento Amazon SNS a un cluster.

Note

Attenendosi alla presente procedura, è anche possibile modificare l'argomento Amazon SNS.

Per aggiungere o modificare l'argomento Amazon SNS per un cluster (console)

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. In Clusters (Cluster), scegliere il cluster al quale aggiungere o di cui modificare un ARN d'argomento Amazon SNS.
3. Scegli Modifica.
4. In Modify Cluster (Modifica cluster) nella sezione Topic for SNS Notification (Argomento per notifica SNS), scegliere l'argomento SNS da aggiungere o scegliere Manual ARN input (Input ARN manuale) e digitare l'ARN dell'argomento Amazon SNS.
5. Scegli Modifica.

Aggiungere un argomento Amazon SNS (AWS CLI)

Per aggiungere o modificare un argomento Amazon SNS per un cluster, usa il AWS CLI comando `update-cluster`

L'esempio di codice riportato di seguito rappresenta l'aggiunta di un ARN d'argomento Amazon SNS a `my-cluster`.

Per Linux, macOS o Unix:

```
aws memorydb update-cluster \  
  --cluster-name my-cluster \  
  --topic-arn arn:aws:sns:us-east-1:123456789012:my-topic
```

```
--sns-topic-arn arn:aws:sns:us-east-1:565419523791:memorydbNotifications
```

Per Windows:

```
aws memorydb update-cluster ^  
  --cluster-name my-cluster ^  
  --sns-topic-arn arn:aws:sns:us-east-1:565419523791:memorydbNotifications
```

Per ulteriori informazioni, consulta [UpdateCluster](#).

Aggiungere un argomento Amazon SNS (API MemoryDB)

Per aggiungere o aggiornare un argomento Amazon SNS per un cluster, richiama l'UpdateClusterazione con i seguenti parametri:

- ClusterName=my-cluster
- SnsTopicArn=arn%3Aaws%3Asns%3Aus-east-1%3A565419523791%3AmemorydbNotifications

Per aggiungere o aggiornare un argomento Amazon SNS per un cluster, chiama l'azioneUpdateCluster.

Per ulteriori informazioni, consulta [UpdateCluster](#).

Attivazione e disattivazione delle notifiche Amazon SNS

È possibile, in base alle proprie esigenze, abilitare o disabilitare le notifiche relative a un cluster. La seguente procedura mostra come disabilitare le notifiche Amazon SNS.

Attivazione e disattivazione delle notifiche Amazon SNS (Console)

Per disabilitare le notifiche di Amazon SNS utilizzando il AWS Management Console

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Scegli il pulsante di opzione a sinistra del cluster per il quale desideri modificare la notifica.
3. Scegli Modifica.
4. In Modify Cluster (Modifica cluster) nella sezione Topic for SNS Notification (Argomento per notifica SNS), scegliere Disable Notifications (Disabilita notifiche).

5. Scegli Modifica.

Abilitazione e disabilitazione delle notifiche Amazon SNS (CLI)AWS

Per disabilitare le notifiche Amazon SNS, occorre utilizzare il comando `update-cluster` con i seguenti parametri:

Per Linux, macOS o Unix:

```
aws memorydb update-cluster \  
  --cluster-name my-cluster \  
  --sns-topic-status inactive
```

Per Windows:

```
aws memorydb update-cluster ^  
  --cluster-name my-cluster ^  
  --sns-topic-status inactive
```

Abilitazione e disabilitazione delle notifiche Amazon SNS (API MemoryDB)

Per disabilitare le notifiche Amazon SNS, occorre chiamare l'operazione `UpdateCluster` con i seguenti parametri:

- `ClusterName=my-cluster`
- `SnsTopicStatus=inactive`

Questa chiamata restituisce un output simile al seguente:

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>
```

Visualizzazione degli eventi di MemoryDB

MemoryDB registra gli eventi relativi ai cluster, ai gruppi di sicurezza e ai gruppi di parametri. Queste informazioni includono la data, l'ora, il nome e tipo di fonte e una descrizione dell'evento. È possibile recuperare facilmente gli eventi dal registro utilizzando la console MemoryDB, il AWS CLI `describe-events` comando o l'azione dell'API MemoryDB. `DescribeEvents`

Le procedure seguenti mostrano come visualizzare tutti gli eventi di MemoryDB delle ultime 24 ore (1440 minuti).

Visualizzazione degli eventi di MemoryDB (Console)

La procedura seguente visualizza gli eventi utilizzando la console MemoryDB.

Per visualizzare gli eventi utilizzando la console MemoryDB

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Nel riquadro di navigazione a sinistra, scegli Eventi.

Viene visualizzata la schermata Eventi che elenca tutti gli eventi disponibili. Ogni riga dell'elenco rappresenta un evento e mostra l'origine dell'evento, il tipo di evento (ad esempio cluster, parameter-group, acl, security-group o subnet group), l'ora GMT dell'evento e la descrizione dell'evento.

La voce Filter (Filtra) consente di specificare se si preferisce visualizzare in elenco tutti gli eventi o solo quelli di un tipo specifico.

Visualizzazione degli eventi MemoryDB (CLI AWS)

Per generare un elenco di eventi MemoryDB utilizzando il, utilizzare il AWS CLI comando. `describe-events` Tramite parametri facoltativi è anche possibile specificare il tipo, l'intervallo di tempo, il numero massimo e altre peculiarità degli eventi da includere nell'elenco.

Il codice seguente elenca fino a 40 eventi del cluster.

```
aws memorydb describe-events --source-type cluster --max-results 40
```

Il codice seguente elenca tutti gli eventi delle ultime 24 ore (1440 minuti).

```
aws memorydb describe-events --duration 1440
```

L'output del comando `describe-events` è simile a quello riportato.

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

Per ulteriori informazioni, tra cui i parametri disponibili e i valori consentiti per tali parametri, consulta [describe-events](#).

Visualizzazione degli eventi MemoryDB (API MemoryDB)

Per generare un elenco di eventi MemoryDB utilizzando l'API MemoryDB, utilizzate l'azione `DescribeEvents`. Tramite parametri facoltativi è anche possibile specificare il tipo, l'intervallo di tempo, il numero massimo e altre peculiarità degli eventi da includere nell'elenco.

Il codice seguente elenca i 40 eventi `-cluster` più recenti.

```
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>
```

Il codice seguente elenca gli eventi del cluster delle ultime 24 ore (1440 minuti).

```
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>
```

Le operazioni descritte in precedenza dovrebbero generare un output simile al seguente.

```
<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>
```

Per ulteriori informazioni, tra cui i parametri disponibili e i valori consentiti per tali parametri, consulta [DescribeEvents](#).

Notifiche di eventi Amazon SNS

MemoryDB può pubblicare messaggi utilizzando Amazon Simple Notification Service (SNS) quando si verificano eventi significativi in un cluster. Questa funzionalità può essere utilizzata per aggiornare gli elenchi dei server sulle macchine client connesse a endpoint di singoli nodi afferenti a un cluster.

Note

Per ulteriori informazioni su Amazon Simple Notification Service (SNS) e relativi prezzi e per i link alla documentazione Amazon SNS, consulta la [Pagina del prodotto Amazon SNS](#).

Le notifiche vengono pubblicate su un Amazon SNS specificato Argomento. Di seguito sono riportati i requisiti delle notifiche:

- È possibile configurare un solo argomento per le notifiche di MemoryDB.
- La AWSL'account che possiede l'argomento Amazon SNS deve corrispondere al proprietario del cluster su cui sono abilitate le notifiche.


Eventi MemoryDB


I seguenti eventi MemoryDB attivano le notifiche Amazon SNS:

Nome evento	Messaggio	Descrizione
Memory DB: Aggiungi nodo completo	"Modified number of nodes from %d to %d"	Un nodo è stato aggiunto al cluster ed è pronto per l'uso.
MemoryDB: AddNodeFailed due to insufficienti	"Failed to modify number of nodes from %d to %d due to insufficient free IP addresses"	Non è stato possibile aggiungere un nodo di nodo per l'insufficienza di indirizzi IP disponibili.
Memory DB: i parametri del cluster sono cambiati	"Updated parameter group for the cluster" In caso di creazione, viene inviato anche il messaggio "Updated to use a ParameterGroup %s"	Uno o più parametri del cluster sono stati modificati.

Nome evento	Messaggio	Descrizione
Memory DB: provisioning del cluster completato	"Cluster created."	Il provisioning di un cluster è completo e i nodi nel cluster sono pronti all'uso.
MemoryDB: il provisioning del cluster non è riuscito a causa di uno stato di rete incompatibile	"Failed to create cluster due to incompatible network state. %s"	È stato effettuato un tentativo di avvio di un nuovo cluster in un cloud privato virtuale (VPC) inesistente.
Memory DB: ripristino cluster fallito	"Restore from %s failed for node %s. %s"	MemoryDB non è stato in grado di popolare il cluster con i dati snapshot di Redis. Ciò potrebbe essere dovuto a un file di snapshot inesistente in Amazon S3 o a autorizzazioni errate per quel file. Lo stato della cluster, sarà <code>restore-failed</code> . È necessario eliminare il cluster e ricominciare. Per ulteriori informazioni, consultare Seminare un nuovo cluster con un'istantanea creata esternamente .
Memory DB: Cluster Scaling completo	"Succeeded applying modification to node type to %s."	Ridimensionamento di un cluster completato.
Memory DB: scalabilità cluster non riuscita	"Failed applying modification to node type to %s."	Operazione di scalabilità verticale sul cluster non riuscita.

Nome evento	Messaggio	Descrizione
Memory DB: gruppo di sicurezza del cluster modificato	"Modified security group for cluster."	<p>Si è verificato uno dei seguenti eventi:</p> <ul style="list-style-type: none">• L'elenco dei gruppi di sicurezza autorizzati per il cluster è stato modificato.• Uno o più nuovi gruppi di sicurezza EC2 sono stati autorizzati nei gruppi di sicurezza associati al cluster.• Uno o più gruppi di sicurezza EC2 sono stati rimossi dai gruppi di sicurezza associati al cluster.

Nome evento	Messaggio	Descrizione
Memory DB: Node Replace Avviato	"Recovering node %s"	<p>MemoryDB ha rilevato che l'host che esegue un nodo è danneggiato o irraggiungibile ed è iniziata la sostituzione di tale nodo.</p> <div data-bbox="1068 493 1507 760" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>La voce DNS relativa al nodo in sostituzione non cambia.</p></div> <p>Nella maggior parte dei casi, non è necessario aggiornar e l'elenco dei server per i client, quando si verifica questo evento. Tuttavia, alcune librerie client potrebbero o smettere di utilizzare il nodo nonostante la sostituzione da parte di MemoryDB dopo la sostituzione da parte di MemoryDB; in tal caso, l'applicazione dovrà aggiornar e l'elenco dei server.</p>

Nome evento	Messaggio	Descrizione
Memory DB: nodo sostituisce completato	"Finished recovery for node %s"	<p>MemoryDB ha rilevato che l'host che esegue un nodo è danneggiato o irraggiungibile ed effettuato la sostituzione di tale nodo.</p> <div data-bbox="1068 493 1510 760" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>La voce DNS relativa al nodo in sostituzione non cambia.</p></div> <p>Nella maggior parte dei casi, non è necessario aggiornar e l'elenco dei server per i client, quando si verifica questo evento. Tuttavia, alcune librerie client potrebbero o smettere di utilizzare il nodo nonostante la sostituzione da parte di MemoryDB dopo la sostituzione da parte di MemoryDB; in tal caso, l'applicazione dovrà aggiornar e l'elenco dei server.</p>
Memory DB: Crea cluster completo	"Cluster created"	Il cluster è stato creato correttamente.

Nome evento	Messaggio	Descrizione
Memory DB: creazione cluster non riuscita	"Failed to create cluster due to unsuccessful creation of its node(s)." e "Deleting all nodes belonging to this cluster."	Il cluster non è stato creato.
Memory DB: Elimina Cluster Completa	"Cluster deleted."	L'eliminazione di un cluster e di tutti i nodi associati è stata completata.
Memory DB: failover completato	"Failover to replica node %s completed"	Il failover su un nodo di replica ha avuto esito positivo.
Memory DB: sostituzione del nodo annullata	"The replacement of node %s which was scheduled during the maintenance window from start time: %s, end time: %s has been canceled"	È stata annullata la sostituzione programmata di un nodo nel cluster.
Memory DB: sostituzione del nodo riprogrammata	"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"	È stata riprogrammata la già prevista sostituzione di un nodo del cluster in una nuova finestra riportata nella notifica. Per informazioni su cosa fare in questa situazione, consulta Sostituzione dei nodi .

Nome evento	Messaggio	Descrizione
Memory DB: sostituzione dei nodi pianificata	"The node %s is scheduled for replacement during the maintenance window from start time: %s to end time: %s"	È stata programmata la sostituzione di un nodo del cluster nella finestra riportata nella notifica. Per informazioni su cosa fare in questa situazione, consulta Sostituzione dei nodi .
Memory DB: rimuovi il nodo completo	"Removed node %s"	Un nodo è stato rimosso dal cluster.
Memory DB: Snapshot completo	"Snapshot %s succeeded for node %s"	Una snapshot è stata completata correttamente.
Memory DB: snapshot fallito	"Snapshot %s failed for node %s"	Una snapshot non è andata a buon fine. Per informazioni dettagliate sulle cause, consulta gli eventi del cluster. Per descrivere l'snapshot, consultare DescribeSnapshots , lo stato sarà failed.

Registrazione delle chiamate API MemoryDB per Redis con AWS CloudTrail

MemoryDB for Redis è integrato con AWS CloudTrail, un servizio che offre un record delle operazioni eseguite da un utente, un ruolo o un AWS servizio in MemoryDB for Redis. CloudTrail acquisisce tutte le chiamate API per MemoryDB for Redis come eventi, incluse le chiamate dalla console MemoryDB for Redis e dal codice al MemoryDB for Redis API. Se si crea un trail, è possibile abilitare la distribuzione continua di CloudTrail eventi in un bucket Amazon S3, inclusi gli eventi per MemoryDB for Redis. Se non configuri un percorso, è comunque possibile visualizzare gli eventi più recenti nella CloudTrail console in Event history (Cronologia eventi). Le informazioni raccolte da consentono CloudTrail di determinare la richiesta effettuata a MemoryDB for Redis, l'indirizzo IP da cui è partita la richiesta, l'autore della richiesta, il momento in cui è stata eseguita e altri dettagli.

Per saperne di più CloudTrail, consulta la [Guida per AWS CloudTrail l'utente](#).

Informazioni su MemoryDB for Redis in CloudTrail

CloudTrail è abilitato sull'AWS account al momento della sua creazione. Quando si verifica un'attività in MemoryDB for Redis, questa viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio in Cronologia eventi. È possibile visualizzare, cercare e scaricare gli eventi recenti nell'account AWS. Per ulteriori informazioni, consulta [Visualizzazione di eventi mediante la cronologia CloudTrail eventi](#).

Per una registrazione continua degli eventi nell'AWS account, inclusi gli eventi per MemoryDB for Redis, crea un trail. Un trail consente di CloudTrail distribuire i file di log in un bucket Amazon S3. Per impostazione di default, quando crei un trail nella console, il trail sarà valido in tutte le regioni. Il trail registra gli eventi di tutte le regioni nella partizione AWS e distribuisce i file di log nel bucket Amazon S3 specificato. Inoltre, è possibile configurare altri AWS servizi per analizzare con maggiore dettaglio e usare i dati raccolti nei CloudTrail log. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Panoramica della creazione di un percorso](#)
- [CloudTrail Servizi e integrazioni supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)
- [Ricezione di file di CloudTrail log da più regioni](#) e [Ricezione di file di CloudTrail log da più regioni e Ricezione di file di log di](#)

Tutte le azioni di MemoryDB for Redis vengono registrate da CloudTrail. Ad esempio, le chiamate a `DescribeClusters` e `UpdateCluster` le azioni generano voci nei file di CloudTrail registro. `CreateCluster`

Ogni evento o voce del log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali utente IAM o root.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro servizio AWS.

Per ulteriori informazioni, consulta [Elemento CloudTrail userIdentity](#).

Comprensione delle voci dei file di registro di MemoryDB per Redis

Un trail è una configurazione che consente la distribuzione di eventi come i file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di log possono contenere una o più voci di log. Un evento rappresenta una singola richiesta da un'origine e include informazioni sull'operazione richiesta, data e ora dell'operazione, parametri della richiesta e così via. CloudTrail I file di log non sono una traccia stack ordinata delle chiamate pubbliche dell'API, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail log di che illustra l'CreateClusteroperazione.

```
{
  "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"
}

```

L'esempio seguente mostra una voce di CloudTrail log di che illustra l'DescribeClustersoperazione. Nota che per tutte le chiamate (eList*) di MemoryDB for Redis DescrivDescribe* e List, laresponseElements sezione viene rimossa e appare comenu11.

```

{
  "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"
}

```

L'esempio seguente mostra una voce di CloudTrail log di che illustraUpdateCluster l'operazione.

```

{
    "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"
}

```

L'esempio seguente mostra una voce di CloudTrail log di che illustra l'CreateUseroperazione. Nota che per le chiamate MemoryDB for Redis che contengono dati sensibili, tali dati verranno cancellati nell' CloudTrail evento corrispondente, come mostrato nellarequestParameters sezione seguente.

```

{
  "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"

```

}

Convalida della conformità per MemoryDB for Redis

I revisori di terze parti valutano la sicurezza e la conformità di MemoryDB for Redis come parte di diversi programmi di conformità. AWS Questo include:

- Payment Card Industry Data Security Standard (PCI DSS). Per ulteriori informazioni, consulta [PCI DSS](#).
- Health Insurance Portability and Accountability Act Business Associate Agreement (HIPAA BAA). Per ulteriori informazioni, consulta [Compliance HIPAA](#).
- System and Organization Controls (SOC) 1, 2 e 3. Per ulteriori informazioni, consulta [SOC](#).
- Programma federale di gestione dei rischi e delle autorizzazioni (FedRAMP) Moderato. Per ulteriori informazioni, vedere [FedRAMP](#).
- ISO/IEC 27001:2013, 27017:2015, 27018:2019 e ISO/IEC 9001:2015. Per ulteriori informazioni, consulta le certificazioni e i servizi [AWSISO](#) e CSA STAR.

Per un elenco di servizi AWS che rientrano nell'ambito di programmi di conformità specifici, consulta [Servizi AWS coperti dal programma di compliance](#).

È possibile scaricare i report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Download di report in AWS Artifact](#).

La vostra responsabilità di conformità quando utilizzate MemoryDB è determinata dalla sensibilità dei vostri dati, dagli obiettivi di conformità della vostra azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- [Security and Compliance Quick Start Guides \(Guide Quick Start Sicurezza e compliance\)](#): queste guide alla distribuzione illustrano considerazioni relative all'architettura e forniscono procedure per la distribuzione di ambienti di base incentrati sulla sicurezza e sulla conformità su AWS.
- [Risorse per la conformità AWS](#): una raccolta di cartelle di lavoro e guide suddivise per settore e area geografica.
- [Evaluating Resources with Rules](#) nella AWS Config Developer Guide: AWS Config valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.

- [AWS Security Hub](#) Questo servizio AWS fornisce una visione completa dello stato di sicurezza all'interno di AWS che consente di verificare la conformità con gli standard e le best practice di sicurezza del settore.
- [AWS Audit Manager](#): questo servizio AWS aiuta a controllare continuamente l'utilizzo di AWS per semplificare la gestione dei rischi e la conformità alle normative e agli standard di settore.

Sicurezza dell'infrastruttura in Amazon MemoryDL

La sicurezza dell'infrastruttura in Amazon MemoryDL

In qualità di servizio gestito, [MemoryDB AWS](#)

Utilizza le chiamate all'API AWS pubblicate da per accedere a MemoryDL per accedere a MemoryDTL per accedere a M I client devono supportare Transport Layer Security (TLS) 1.2 o versioni successive. È consigliabile TLLTLTLTLTLTLTL I client devono, inoltre, supportare le suite di cifratura con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate tramite un ID chiave di accesso e una chiave di accesso segreta associata a un principal IAM. In alternativa, è possibile utilizzare [AWS Security Token Service](#) (AWS STS) per generare le credenziali di sicurezza temporanee per firmare le richieste.

Riservatezza del traffico Internet

MemoryDB for Redis utilizza le seguenti tecniche per rendere sicuri i dati e proteggerli dagli accessi non autorizzati:

- [MemoryDB e Amazon VPC](#) spiega il tipo di gruppo di sicurezza necessario per l'installazione.
- [API MemoryDB per l'API Redis ed endpoint VPC di interfaccia \(AWS PrivateLink\)](#) consente di stabilire una connessione privata tra gli endpoint VPC e l'API MemoryDB for Redis.
- [Gestione delle identità e degli accessi in MemoryDB per Redis](#) per concedere e limitare le operazioni di utenti, gruppi e ruoli.

MemoryDB e Amazon VPC

Il servizio Virtual Private Cloud (Amazon VPC) di Amazon definisce una rete virtuale che ricorda molto un data center tradizionale. Quando vuoi configurare un cloud privato virtuale (VPC) con Amazon VPC, puoi selezionare l'intervallo di indirizzi IP, creare sottoreti e configurare tabelle di routing, gateway di rete e impostazioni di sicurezza. Puoi anche aggiungere un cluster alla rete virtuale e controllare l'accesso al cluster usando i gruppi di sicurezza di Amazon VPC.

Questa sezione spiega come configurare manualmente un cluster di MemoryDB in un VPC. Queste informazioni sono rivolte agli utenti che desiderano comprendere come interagiscono MemoryDB e Amazon VPC.

Argomenti

- [Informazioni su MemoryDB e VPC](#)
- [Modelli di accesso a un cluster MemoryDB in una Amazon VPC](#)
- [Creazione di un virtual private cloud \(VPC\).](#)

Informazioni su MemoryDB e VPC

MemoryDB è completamente integrato con Amazon VPC. Per gli utenti di MemoryDB, questo significa che:

- MemoryDB avvia sempre il cluster in un VPC.
- Se non hai ancora dimestichezza con AWS, verrà creato automaticamente un VPC predefinito.
- Se disponi di un VPC predefinito e non specifichi una sottorete all'avvio di un cluster, il cluster si avvia nel Amazon VPC predefinito.

Per ulteriori informazioni, consulta la sezione relativa al [rilevamento delle piattaforme supportate e di un eventuale VPC di default](#).

Con Amazon VPC puoi creare una rete virtuale nella AWS Cloud che ricorda molto un data center tradizionale. Puoi configurare il VPC, inclusi la selezione degli intervalli di indirizzi IP, la creazione di sottoreti, la configurazione di tabelle di routing, i gateway di rete, nonché le impostazioni di sicurezza.

MemoryDB gestisce gli aggiornamenti software, le patch, il rilevamento dell'errore e il ripristino.

Panoramica di MemoryDB in un VPC

1

Un VPC è una parte isolata dell'AWS Cloud a cui è stato assegnato il proprio blocco di indirizzi IP.

2

Un gateway Internet collega il tuo VPC direttamente a Internet e fornisce l'accesso ad altre risorse AWS, ad esempio Amazon Simple Storage Service (Amazon S3), che sono in esecuzione all'esterno del VPC.

3

Una sottorete Amazon VPC è un segmento dell'intervallo di indirizzi IP di un VPC in cui è possibile isolare le risorse AWS in base alle esigenze operative e di sicurezza.

4

Una tabella di routing nel VPC instrada il traffico di rete tra la sottorete e Internet. Amazon VPC ha un router implicito.

5

Un gruppo di sicurezza di Amazon VPC controlla il traffico in entrata e in uscita per i cluster di MemoryDB e le istanze Amazon EC2.

6

È possibile avviare un cluster di MemoryDB nella sottorete. I nodi hanno indirizzi IP privati dall'intervallo di indirizzi della sottorete.

7

Puoi anche avviare le istanze Amazon EC2 nella sottorete. Ogni istanza Amazon EC2 ha un indirizzo IP privato dall'intervallo di indirizzi della sottorete. L'istanza Amazon EC2 può connettersi a un qualsiasi nodo nella stessa sottorete.

8

Affinché un'istanza Amazon EC2 nella VPC sia raggiungibile da Internet, è necessario assegnare un indirizzo pubblico, statico, chiamato indirizzo IP elastico all'istanza.

Prerequisiti

Per creare un cluster di MemoryDB all'interno di un VPC, il VPC deve soddisfare i seguenti requisiti:

- Il VPC deve consentire istanze Amazon EC2 non dedicate. Non puoi utilizzare MemoryDB in un VPC configurato per una locazione dell'istanza dedicata.
- È necessario definire un gruppo di sottoreti per il VPC. MemoryDB utilizza quel gruppo di sottoreti per selezionare una sottorete e gli indirizzi IP all'interno di quella sottorete da associare ai nodi.
- È necessario definire un gruppo di sicurezza per il VPC; in alternativa, è possibile utilizzare il valore predefinito fornito.
- I blocchi CIDR per ogni sottorete devono essere abbastanza grandi da fornire indirizzi IP di riserva affinché MemoryDB possa utilizzarli durante le attività di manutenzione.

Routing e sicurezza

Puoi configurare l'instradamento nel VPC per controllare i punti di affluenza del traffico (ad esempio, il gateway Internet o il gateway privato virtuale). Con un gateway internet, il tuo VPC ha accesso diretto ad altri AWS risorse che non sono in esecuzione nel tuo VPC. Se scegli di avere solo un gateway privato virtuale con una connessione alla rete locale della tua organizzazione, puoi instradare il traffico Internet attraverso la VPN e utilizzare il firewall e le policy di sicurezza locali per controllare i punti di uscita. In tal caso, incorri in costi aggiuntivi per la larghezza di banda quando accedi AWS risorse su Internet.

Puoi utilizzare i gruppi di sicurezza di Amazon VPC per contribuire a proteggere i cluster di MemoryDB e le istanze Amazon EC2 nella Amazon VPC. I gruppi di sicurezza operano come un firewall a livello di istanza, non di sottorete.

Note

Consigliamo di utilizzare i nomi DNS per effettuare il collegamento ai nodi, poiché l'indirizzo IP sottostante può variare nel tempo.

Documentazione Amazon VPC

Amazon VPC dispone della propria documentazione che descrive come creare e usare l'Amazon VPC. La tabella seguente mostra dove trovare informazioni nelle guide di Amazon VPC.

Descrizione	Documentazione
Come iniziare a usare Amazon VPC	Nozioni di base su Amazon VPC
Come usare Amazon VPC tramite la AWS Management Console	Guida per l'utente di Amazon VPC
Descrizioni complete di tutti i comandi di Amazon VPC	Guida di riferimento alla riga di comando di Amazon EC2 (i comandi di Amazon VPC si trovano nel riferimento Amazon EC2)
Descrizioni complete di operazioni API, tipi di dati ed errori di Amazon VPC	Guida di riferimento all'API di Amazon EC2 (le operazioni dell'API di Amazon VPC si trovano nel riferimento Amazon EC2)
Informazioni per l'amministratore di reti che deve configurare il gateway di una connessione VPN IPsec opzionale presso la sede dell'utente	Cos'è AWS Site-to-Site VPN?

Per informazioni più dettagliate su Amazon Virtual Private Cloud, consulta [Amazon Virtual Private Cloud](#).

Modelli di accesso a un cluster MemoryDB in una Amazon VPC

MemoryDB per Redis supporta i seguenti scenari per accedere a un cluster in una Amazon VPC:

Indice

- [Accesso a un cluster MemoryDB quando tale cluster e l'istanza Amazon EC2 sono nello stesso Amazon VPC](#)
- [Accesso a un cluster MemoryDB quando tale cluster e l'istanza Amazon EC2 sono in differenti Amazon VPC](#)
 - [Accesso a un cluster MemoryDB quando tale cluster e l'istanza Amazon EC2 sono in differenti Amazon VPC nella stessa regione](#)
 - [Uso del Transit Gateway](#)
 - [Accesso a un cluster MemoryDB quando tale cluster e l'istanza Amazon EC2 sono in differenti Amazon VPC in regioni differenti](#)
 - [Utilizzo di VPC di transito](#)
- [Accesso a un cluster MemoryDB da un'applicazione in esecuzione nel data center di un cliente](#)
 - [Accesso a un cluster MemoryDB da un'applicazione in esecuzione nel data center di un cliente tramite connessione VPN](#)
 - [Accesso a un cluster MemoryDB da un'applicazione in esecuzione nel data center di un cliente tramite Direct Connect](#)

Accesso a un cluster MemoryDB quando tale cluster e l'istanza Amazon EC2 sono nello stesso Amazon VPC

Il caso d'uso più comune è quando un'applicazione distribuita su un'istanza EC2 deve connettersi a un cluster nello stesso VPC.

Il modo più semplice per gestire l'accesso tra istanze EC2 e istanze database nello stesso VPC consiste nel fare quanto segue:

1. Creare un gruppo di sicurezza VPC per il cluster. Questo gruppo di sicurezza può essere utilizzato per limitare l'accesso ai cluster. Per questo gruppo di sicurezza è ad esempio possibile creare una regola personalizzata che consenta l'accesso TCP tramite la porta assegnata al cluster al momento della creazione e un indirizzo IP che verrà utilizzato per accedere al cluster.

La porta predefinita per i cluster MemoryDB è 6379.

2. Creare un gruppo di sicurezza VPC per le istanze EC2 (server Web e di applicazioni). Questo gruppo di sicurezza può, se necessario, consentire l'accesso all'istanza EC2 da Internet tramite la tabella di routing del VPC. Ad esempio, può impostare regole in questo gruppo di sicurezza per consentire l'accesso TCP all'istanza EC2 sulla porta 22.
3. Creare regole personalizzate nel gruppo di sicurezza per il cluster che consentano connessioni dal gruppo di sicurezza creato per le istanze EC2. Ciò consente a qualsiasi membro del gruppo di sicurezza di accedere ai cluster.

Per creare in un gruppo di sicurezza VPC una regola che consenta connessioni da un altro gruppo di sicurezza

1. Accedere alla Console di gestione AWS e aprire la console di Amazon VPC all'indirizzo <https://console.aws.amazon.com/vpc>.
2. Nel riquadro di navigazione a sinistra, scegli Security Groups (Gruppi di sicurezza).
3. Selezionare o creare un gruppo di sicurezza da utilizzare per i cluster. In Regole in entrata, scegliere Modifica regole in entrata e quindi Aggiungi regola. Tale gruppo di sicurezza consentirà di accedere ai membri di un altro gruppo di sicurezza.
4. In Type (Tipo) scegliere Custom TCP Rule (Regola TCP personalizzata).
 - a. Per Port Range (Intervallo porte) specificare la porta utilizzata alla creazione del cluster.

La porta predefinita per i cluster MemoryDB è 6379.
 - b. Nella casella Source (fonte) iniziare a digitare l'ID del gruppo di sicurezza. Nell'elenco selezionare il gruppo di sicurezza da utilizzare per le istanze Amazon EC2.
5. Scegliere Save (Salva) al termine.

Accesso a un cluster MemoryDB quando tale cluster e l'istanza Amazon EC2 sono in differenti Amazon VPC

Quando il cluster si trova in un VPC diverso dall'istanza EC2 che utilizzi per accedervi, è possibile accedere al cluster in diversi modi. Se il cluster e l'istanza EC2 si trovano in VPC differenti, ma nella stessa regione, puoi utilizzare il peering VPC. Se il cluster e l'istanza EC2 si trovano in regioni differenti, puoi creare una connessione VPN tra le regioni.

Argomenti

- [Accesso a un cluster MemoryDB quando tale cluster e l'istanza Amazon EC2 sono in differenti Amazon VPC nella stessa regione](#)
- [Accesso a un cluster MemoryDB quando tale cluster e l'istanza Amazon EC2 sono in differenti Amazon VPC in regioni differenti](#)

Accesso a un cluster MemoryDB quando tale cluster e l'istanza Amazon EC2 sono in differenti Amazon VPC nella stessa regione

Cluster a cui si accede tramite una istanza Amazon EC2 in una Amazon VPC differente nella stessa regione - Connessione peering VPC

Una connessione di peering di VPC è una connessione di rete tra due VPC che consentono di instradare il traffico tra loro utilizzando degli indirizzi IP privati. Le istanze in uno qualsiasi dei VPC possono comunicare tra loro come se fossero nella stessa rete. Puoi anche creare una connessione peering VPC tra le tue Amazon VPC o con una Amazon VPC in un altro account AWS, all'interno di una singola regione. Per ulteriori informazioni sul peering VPC di Amazon consulta la [documentazione relativa alla VPC](#).

Per accedere a un cluster in una Amazon VPC differente sul peering

1. Assicurarsi che i due VPC non abbiano un intervallo IP sovrapposto, altrimenti non si potrà sottoporli a peering.
2. Eseguire il peering dei due VPC. Per ulteriori informazioni, consulta [Creare e accettare una connessione peering VPC di Amazon](#).
3. Aggiornare la tabella di routing. Per ulteriori informazioni, consulta [Aggiornamento delle tabelle di routing per una connessione peering VPC](#).
4. Modificare il gruppo di sicurezza del cluster MemoryDB per consentire la connessione in entrata dal gruppo di sicurezza dell'applicazione nel VPC in peering. Per ulteriori informazioni, vedi l'argomento relativo ai [gruppi di sicurezza nel VPC in peering](#).

L'accesso a un cluster in una connessione peering implicherà ulteriori costi di trasferimento dei dati.

Uso del Transit Gateway

Un Transit Gateway consente di collegare VPC e connessioni VPN nella stessa regione AWS e routing di traffico tra loro. Un Transit Gateway funziona su tutti gli account AWS ed è possibile utilizzare AWS Resource Access Manager di AWS per condividere il Transit Gateway con altri account. Dopo aver condiviso un gateway di transito con un altro account AWS, il proprietario dell'account può collegare i propri VPC al gateway di transito. Un utente di uno qualsiasi degli account può eliminare il collegamento in qualsiasi momento.

È possibile abilitare il multicast in un gateway di transito, quindi creare un dominio del gateway di transito multicast che consenta l'invio del traffico multicast dall'fonte multicast ai membri del gruppo multicast tramite allegati VPC associati al dominio.

È inoltre possibile creare un allegato di connessione di peering tra gateway di transito in diverse regioni AWS. In questo modo è possibile instradare il traffico tra gli allegati dei gateway di transito in diverse regioni.

Per ulteriori informazioni, consulta [Gateway di transito](#).

Accesso a un cluster MemoryDB quando tale cluster e l'istanza Amazon EC2 sono in differenti Amazon VPC in regioni differenti

Utilizzo di VPC di transito

Un'alternativa all'utilizzo del peering VPC, un'altra strategia comune per connettere più VPC situati in differenti zone geografiche e reti remote consiste nel creare un VPC di transito utilizzabile come centro di transito di rete globale. Un VPC di transito semplifica la gestione della rete e riduce al minimo il numero di connessioni necessarie per connettersi a più VPC e reti remote. Questo tipo di progettazione può consentirti di risparmiare tempo, limitare il lavoro necessario e ridurre i costi, in quanto è praticamente implementato senza la spesa in genere necessaria per stabilire una presenza fisica in un hub di transito di co-location o per distribuire un'apparecchiatura di rete fisica.

Connessione tra VPC differenti in diverse regioni

Una volta che viene definito l'Amazon VPC di transito, un'applicazione distribuita in un VPC «spoke» in una regione può essere connessa a un cluster MemoryDB in un VPC «spoke» all'interno di un'altra regione.

Per accedere a un cluster in un VPC differente in una regione AWS differente

1. Distribuire una soluzione VPC di transito. Per ulteriori informazioni, consulta [AWSTransit Gateway](#).
2. Aggiornare le tabelle di routing VPC nell'app e nei VPC per instradare il traffico tramite il gateway privato virtuale (VGW) e l'appliance VPN. Nel caso di un routing dinamico con Border Gateway Protocol (BGP) le route possono essere automaticamente propagate.
3. Modificare il gruppo di sicurezza del cluster MemoryDB per consentire la connessione in entrata dall'intervallo IP delle istanze dell'applicazione. In questo scenario, non è possibile fare riferimento al gruppo di sicurezza del server di applicazioni.

L'accesso a un cluster tra regioni introdurrà latenze di rete e ulteriori costi di trasferimento dei dati tra regioni.

Accesso a un cluster MemoryDB da un'applicazione in esecuzione nel data center di un cliente

Un altro possibile scenario è rappresentato da un'architettura ibrida in cui i client o le applicazioni nel data center del cliente potrebbero dover accedere a un cluster MemoryDB nel VPC. Anche questo scenario è supportato purché sia disponibile una connessione tra VPC e data center dei clienti tramite VPN o Direct Connect.

Argomenti

- [Accesso a un cluster MemoryDB da un'applicazione in esecuzione nel data center di un cliente tramite connessione VPN](#)
- [Accesso a un cluster MemoryDB da un'applicazione in esecuzione nel data center di un cliente tramite Direct Connect](#)

Accesso a un cluster MemoryDB da un'applicazione in esecuzione nel data center di un cliente tramite connessione VPN

Connessione a MemoryDB dal data center tramite una VPN

Per accedere a un cluster in un VPC da un'applicazione locale su una connessione VPN

1. Stabilire una connessione VPN aggiungendo un gateway privato virtuale hardware al proprio VPC. Per ulteriori informazioni, consulta [Aggiunta di un gateway privato virtuale hardware al proprio VPC](#).
2. Aggiornare la tabella di routing VPC per la sottorete in cui viene distribuito il cluster MemoryDB per consentire il traffico dal server di applicazioni locale. Nel caso di un routing dinamico con BGP le route possono essere automaticamente propagate.
3. Modificare il gruppo di sicurezza del cluster MemoryDB per consentire la connessione in entrata dai server di applicazioni locali.

L'accesso a un cluster su una connessione VPN introdurrà latenze di rete e ulteriori costi di trasferimento dei dati.

Accesso a un cluster MemoryDB da un'applicazione in esecuzione nel data center di un cliente tramite Direct Connect

Connessione a MemoryDB dal data center tramite Direct Connect

Per accedere a un cluster MemoryDB da un'applicazione in esecuzione in una rete tramite Direct Connect

1. Stabilire una connessione Direct Connect. Per ulteriori informazioni, consulta [Nozioni di base su AWS Direct Connect](#).
2. Modificare il gruppo di sicurezza del cluster MemoryDB per consentire la connessione in entrata dai server di applicazioni locali.

L'accesso a un cluster su una connessione DX può introdurre latenze di rete e ulteriori costi di trasferimento dei dati.

Creazione di un virtual private cloud (VPC).

In questo esempio, si crea un cloud privato virtuale (VPC) basato sul servizio Amazon VPC con una sottorete privata per ogni zona di disponibilità.

Creazione di un VPC (Console)

Per creare un cluster di MemoryDB all'interno di un Amazon Virtual Private Cloud

1. Accedere alla Console di gestione AWS e aprire la console Amazon VPC all'indirizzo <https://console.aws.amazon.com/vpc/>.
2. Nel pannello di controllo VPC, scegli Creazione di un VPC.
3. Per Resources to create (Risorse da creare), scegli VPC and more (VPC e altro).
4. Per Number of Availability Zones (AZs) (Numero di zone di disponibilità [AZ]), scegli il numero di zone di disponibilità in cui desideri avviare le sottoreti.
5. Per Number of public subnets (Numero di sottoreti pubbliche), scegli il numero di sottoreti pubbliche che vuoi aggiungere al tuo VPC.
6. Per Number of private subnets (Numero di sottoreti private), scegli il numero di sottoreti private che vuoi aggiungere al tuo VPC.

Tip

Prendi nota degli identificatori di sottorete e di quale sia quello pubblico e quale quello privato. Queste informazioni saranno necessarie in seguito quando avvierai i cluster e aggiungerai un'istanza Amazon EC2 al tuo Amazon VPC.

7. Creare un gruppo di sicurezza Amazon VPC Utilizzerai questo gruppo per il cluster e l'istanza Amazon EC2.
 - a. Nel riquadro di navigazione a sinistra della AWS Management Console, scegli Security Groups (Gruppi di sicurezza).
 - b. Scegli Crea gruppo di sicurezza.
 - c. Inserire un nome e una descrizione per il gruppo di sicurezza nelle caselle corrispondenti. Per VPC, seleziona l'identificativo del VPC.
 - d. Dopo aver selezionato tutte le impostazioni che desideri, scegliere Yes, Create (Crea).
8. Definire una regola di ingresso di rete per il gruppo di sicurezza. Questa regola ti consente di collegarti alla tua istanza Amazon EC2 mediante l'SSH (Secure Shell).

- a. Nel riquadro di navigazione a sinistra, scegli Security Groups (Gruppi di sicurezza).
- b. Occorre trovare il gruppo di sicurezza nell'elenco, quindi selezionarlo.
- c. In Security groups (Gruppi di sicurezza), scegliere la scheda Inbound (In entrata). Nella casella Create a new rule (Crea una nuova regola), scegliere SSH, quindi Add Rule (Aggiungi regola).

Impostare i seguenti valori per la nuova regola in entrata per consentire l'accesso HTTP:

- Type: HTTP
- Origine: 0.0.0.0/0

- d. Impostare i seguenti valori per la nuova regola in entrata per consentire l'accesso HTTP:

- Type: HTTP
- Origine: 0.0.0.0/0

Scegliere Apply Rule Changes (Applica modifiche della regola).

Ora sei pronto per creare un [Gruppo di sottorete](#) e [creare un cluster](#) nel VPC.

Sottoreti e gruppi di sottoreti

Un gruppo di sottoreti è una raccolta di sottoreti (generalmente private) che è possibile designare per i cluster in esecuzione in un ambiente Amazon Virtual Private Cloud (VPC)

Quando crei un cluster in un Amazon VPC, puoi specificare un gruppo di sottoreti o utilizzare quello predefinito fornito. MemoryDB utilizza quel gruppo di sottoreti per scegliere una sottorete e gli indirizzi IP all'interno di quella sottorete da associare ai nodi.

Questa sezione spiega come creare e sfruttare sottoreti e gruppi di sottoreti per gestire l'accesso alle risorse di MemoryDB.

Per ulteriori informazioni sull'utilizzo dei gruppi di sottoreti in un ambiente Amazon VPC, consulta [Passaggio 2: autorizzare l'accesso al cluster](#).

ID MemoryDB AZ supportati

Nome regione/Regione	ID AZ supportati		
Stati Uniti orientali (Ohio) us-east-2	use2-az1, use2-az2, use2-az3		
Stati Uniti orientali (Virginia settentrionale) us-east-1	use1-az2, use1-az4, use1-az6		
Regione Stati Uniti occidentali (California settentrionale) us-west-1	usw1-az1, usw1-az2, usw1-az3		
Stati Uniti occidentali (Oregon) us-west-2	usw2-az1, usw2-az2, usw2-az3		

Nome regione/Regione	ID AZ supportati		
Regione Canada (Centrale) ca-central-1	cac1-az1, cac1-az2, cac1-az4		
Regione Asia Pacifico (Hong Kong) ap-east-1	ape1-az1, ape1-az2, ape1-az3		
Regione Asia Pacifico (Mumbai) ap-south-1	aps1-az1, aps1-az2, aps1-az3		
Regione Asia Pacifico (Tokyo) ap-northeast-1	apne1-az1, apne1-az2, apne1-az4		
Asia Pacific (Seoul) Region ap-northeast-2	apne2-az1, apne2-az2, apne2-az3		
Regione Asia Pacifico (Singapore) ap-southeast-1	apse1-az1, apse1-az2, apse1-az3		
Regione Asia Pacifico (Sydney) ap-southeast-2	apse2-az1, apse2-az2, apse2-az3		

Nome regione/Regione	ID AZ supportati		
Regione Europa (Francoforte) eu-central-1	euc1-az1, euc1-az2, euc1-az3		
Regione Europa (Irlanda) eu-west-1	euw1-az1, euw1-az2, euw1-az3		
Regione Europa (Londra) eu-west-2	euw2-az1, euw2-az2, euw2-az3		
Regione UE (Parigi) eu-west-3	euw3-az1, euw3-az2, euw3-az3		
Regione Europa (Stoccolma) eu-north-1	eun1-az1, eun1-az2, eun1-az3		
Regione Europa (Milano) eu-south-1	eus1-az1, eus1-az2, eus1-az3		
Regione Sud America (San Paolo) sa-east-1	sae1-az1, sae1-az2, sae1-az3		
Regione Cina (Pechino) cn-north-1	cnn1-az1, cnn1-az2		

Nome regione/Regione	ID AZ supportati		
Regione Cina (Ningxia) cn-northwest-1	cnw1-az1, cnw1-az2, cnw1-az3		

Argomenti

- [Creazione di un gruppo di sottoreti](#)
- [Aggiornamento di un gruppo di sottoreti](#)
- [Visualizzazione dei dettagli del gruppo di sottoreti](#)
- [Eliminazione di un gruppo di sottoreti](#)

Creazione di un gruppo di sottoreti

Quando si crea un nuovo gruppo di sottoreti, tieni presente il numero di indirizzi IP disponibili. Se la sottorete ha un numero molto ridotto di indirizzi IP gratuiti, potresti avere delle limitazioni sul numero di nodi aggiuntivi da aggiungere al cluster. Per risolvere questo problema, è possibile assegnare una o più sottoreti a un gruppo di sottoreti in modo da avere un numero sufficiente di indirizzi IP nella zona di disponibilità del cluster. Dopodiché, è possibile aggiungere ulteriori nodi al cluster.

Le seguenti procedure mostrano come creare un gruppo di sottoreti chiamato `mysubnetgroup` (console) AWS CLI, e l'API MemoryDB.

Creazione di un gruppo di sottoreti (Console)

La procedura seguente mostra come creare un gruppo di sottoreti (console).

Per creare un gruppo di sottoreti (Console)

1. [Accedere alla console di AWS gestione e aprire la console di MemoryDB all'indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. Nel riquadro di navigazione a sinistra, scegli Subnet Groups.
3. Scegliere Create Subnet Group (Crea gruppo di sottoreti).
4. Nella pagina Crea gruppo di sottoreti, procedi come segue:
 - a. Nella casella Name (Nome), digitare un nome per il gruppo di sottoreti.

I vincoli di denominazione dei cluster sono i seguenti:
 - Devono contenere da 1 a 40 caratteri alfanumerici o trattini.
 - Devono iniziare con una lettera.
 - Non possono contenere due trattini consecutivi.
 - Non possono terminare con un trattino.
 - b. Nella casella Description (Descrizione), digitare una descrizione per il gruppo di sottoreti.
 - c. Nella casella VPC ID (ID VPC), scegliere l'Amazon VPC creato. Se non ne hai ancora creato uno, scegli il pulsante Crea VPC e segui i passaggi per crearne uno.
 - d. In Sottoreti selezionate, scegli la zona di disponibilità e l'ID della tua sottorete privata, quindi scegli.
5. Per i tag, puoi facoltativamente applicare tag per cercare e filtrare le sottoreti o tenere traccia dei costi. AWS

6. Dopo aver selezionato tutte le impostazioni desiderate, scegli Crea.
7. Nel messaggio di conferma visualizzato, scegliere Close (Chiudi).

Il nuovo gruppo di sottoreti viene visualizzato nell'elenco dei gruppi di sottoreti della console MemoryDB. Nella parte in basso della finestra puoi scegliere il gruppo di sottoreti per visualizzare i dettagli, ad esempio tutte le sottoreti associate a tale gruppo.

Creazione di un gruppo di sottoreti (AWS CLI)

Al prompt dei comandi, utilizzare il comando `create-subnet-group` per creare un gruppo di sottoreti.

Per Linux, macOS o Unix:

```
aws memorydb create-subnet-group \  
  --subnet-group-name mysubnetgroup \  
  --description "Testing" \  
  --subnet-ids subnet-53df9c3a
```

Per Windows:

```
aws memorydb create-subnet-group ^  
  --subnet-group-name mysubnetgroup ^  
  --description "Testing" ^  
  --subnet-ids subnet-53df9c3a
```

Questo comando dovrebbe generare un output simile al seguente:

```
{  
  "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"
    }
}
```

Per ulteriori informazioni, consultate l' AWS CLI argomento. [create-subnet-group](#)

Creazione di un gruppo di sottoreti (API MemoryDB)

Utilizzando l'API MemoryDB, chiamate `CreateSubnetGroup` con i seguenti parametri:

- `SubnetGroupName`=*mysubnetgroup*
- `Description`=*Testing*
- `SubnetIds.member.1`=*subnet-53df9c3a*

Aggiornamento di un gruppo di sottoreti

È possibile aggiornare la descrizione di un gruppo di sottoreti o modificare l'elenco degli ID di sottorete associati al gruppo di sottoreti. Non è possibile eliminare un ID di sottorete da un gruppo se un cluster utilizza attualmente quella sottorete.

Le procedure seguenti mostrano come aggiornare un gruppo di sottoreti.

Aggiornamento dei gruppi di sottoreti (Console)

Per aggiornare un gruppo di sottoreti

1. [Accedere AWS Management Console e aprire la console MemoryDB for Redis all'indirizzo `https://console.aws.amazon.com/memorydb/`.](https://console.aws.amazon.com/memorydb/)
2. Nel riquadro di navigazione a sinistra, scegli Subnet Groups.
3. Nell'elenco dei gruppi di sottoreti, scegliere quello che si desidera modificare.
4. I campi Nome, VPCId e Descrizione non sono modificabili.
5. Nella sezione Sottoreti selezionate, fai clic su Gestisci per apportare le modifiche alle zone di disponibilità necessarie per le sottoreti. Per salvare le modifiche, scegliere Save (Salva).

Aggiornamento dei gruppi di sottoreti (AWS CLI)

Al prompt dei comandi, utilizzate il comando `update-subnet-group` per aggiornare un gruppo di sottoreti.

Per Linux, macOS o Unix:

```
aws memorydb update-subnet-group \  
  --subnet-group-name mysubnetgroup \  
  --description "New description" \  
  --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Per Windows:

```
aws memorydb update-subnet-group ^  
  --subnet-group-name mysubnetgroup ^  
  --description "New description" ^  
  --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Questo comando dovrebbe generare un output simile al seguente:

```
{
  "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",
  }
}
```

Per ulteriori informazioni, consultate l' AWS CLI argomento. [update-subnet-group](#)

Aggiornamento dei gruppi di sottoreti (API MemoryDB)

Utilizzando l'API MemoryDB, chiamate UpdateSubnetGroup con i seguenti parametri:

- SubnetGroupName=*mysubnetgroup*
- Qualsiasi altro parametro di cui si desidera modificare i valori. Questo esempio utilizza Description=*New%20description* per modificare la descrizione del gruppo di sottoreti.

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

Quando si crea un gruppo di sottoreti, prendere nota del numero di indirizzi IP disponibili. Se la sottorete ha un numero molto ridotto di indirizzi IP gratuiti, potresti avere delle limitazioni sul numero di nodi aggiuntivi da aggiungere al cluster. Per risolvere questo problema, è possibile assegnare una o più sottoreti a un gruppo di sottoreti in modo da avere un numero sufficiente di indirizzi IP nella zona di disponibilità del cluster. Dopodiché, è possibile aggiungere ulteriori nodi al cluster.

Visualizzazione dei dettagli del gruppo di sottoreti

Le procedure seguenti mostrano come visualizzare i dettagli di un gruppo di sottoreti.

Visualizzazione dei dettagli dei gruppi di sottoreti (console)

Per visualizzare i dettagli di un gruppo di sottoreti (Console)

1. [Accedere AWS Management Console e aprire la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/](https://console.aws.amazon.com/memorydb/).
2. Nel riquadro di navigazione a sinistra, scegli Subnet Groups.
3. Nella pagina Gruppi di sottoreti, scegli il gruppo di sottoreti in Nome o inserisci il nome del gruppo di sottoreti nella barra di ricerca.
4. Nella pagina Gruppi di sottoreti, scegli il gruppo di sottoreti in Nome o inserisci il nome del gruppo di sottoreti nella barra di ricerca.
5. Nelle impostazioni del gruppo di sottorete puoi visualizzare il nome, la descrizione, l'ID VPC e l'Amazon Resource Name (ARN) del gruppo di sottoreti.

6. In Subnet puoi visualizzare le zone di disponibilità, gli ID di sottorete e i blocchi CIDR del gruppo di sottoreti
7. In Tag è possibile visualizzare tutti i tag associati al gruppo di sottoreti.

Visualizzazione dei dettagli dei gruppi di sottoreti (AWS CLI)

Al prompt dei comandi, utilizzate il comando `describe-subnet-groups` per visualizzare i dettagli di un gruppo di sottoreti specificato.

Per Linux, macOS o Unix:

```
aws memorydb describe-subnet-groups \  
  --subnet-group-name mysubnetgroup
```

Per Windows:

```
aws memorydb describe-subnet-groups ^  
  --subnet-group-name mysubnetgroup
```

Questo comando dovrebbe generare un output simile al seguente:

```
{  
  "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"
}
]
}

```

Per visualizzare i dettagli su tutti i gruppi di sottoreti, utilizzate lo stesso comando ma senza specificare il nome del gruppo di sottorete.

```
aws memorydb describe-subnet-groups
```

Per ulteriori informazioni, consultate l'argomento. AWS CLI [describe-subnet-groups](#)

Visualizzazione dei gruppi di sottoreti (API MemoryDB)

Utilizzando l'API MemoryDB, chiamate DescribeSubnetGroups con i seguenti parametri:

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`

Eliminazione di un gruppo di sottoreti

Se ritieni che il gruppo di sottoreti non sia più necessario, puoi eliminarlo. Non è possibile eliminare un gruppo di sottoreti se è attualmente utilizzato da un cluster. Non è inoltre possibile eliminare un gruppo di sottoreti in un cluster con la funzione Multi-AZ abilitato se tale cluster non contiene più di due sottoreti. È necessario prima deselezionare Multi-AZ e quindi eliminare la sottorete.

Le procedure seguenti mostrano come eliminare un gruppo di sottoreti.

Eliminazione di un gruppo di sottoreti (Console)

Per eliminare un gruppo di sottoreti

1. [Accedi AWS Management Console e apri la console MemoryDB for Redis all'indirizzo https://console.aws.amazon.com/memorydb/.](https://console.aws.amazon.com/memorydb/)
2. Nel riquadro di navigazione a sinistra, scegli Subnet Groups.
3. Nell'elenco dei gruppi di sottoreti, scegli quello che desideri eliminare, scegli Azioni e quindi scegli Elimina.

Note

Non è possibile eliminare un gruppo di sottoreti predefinito o associato a qualsiasi cluster.

4. Verrà visualizzata la schermata di conferma dell'eliminazione dei gruppi di sottorete.
5. Per eliminare il gruppo di sottoreti, inseriscilo `delete` e nella casella di testo di conferma. Per mantenere il gruppo di sottoreti, scegliere `Cancel` (Annulla).

Eliminazione di un gruppo di sottoreti (CLI AWS)

Utilizzando AWS CLI, chiamate il comando `delete-subnet-group` con il seguente parametro:

- `--subnet-group-name mysubnetgroup`

Per Linux, macOS o Unix:

```
aws memorydb delete-subnet-group \  
  --subnet-group-name mysubnetgroup
```

Per Windows:

```
aws memorydb delete-subnet-group ^  
  --subnet-group-name mysubnetgroup
```

Per ulteriori informazioni, consulta l' AWS CLI argomento [delete-subnet-group](#).

Eliminazione di un gruppo di sottoreti (API MemoryDB)

Utilizzando l'API MemoryDB, chiamate con il seguente parametro: DeleteSubnetGroup

- 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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consultate l'argomento API MemoryDB. [DeleteSubnetGroup](#)

API MemoryDB per l'API Redis ed endpoint VPC di interfaccia (AWS PrivateLink)

È possibile stabilire una connessione privata tra VPC e Amazon Endpoint API MemoryDB per Redis creando un endpoint VPC dell'interfaccia. Endpoint di interfaccia sono alimentati da [AWS PrivateLink](#). AWS PrivateLink permette di accedere privatamente alle operazioni API Redis senza un gateway Internet, un dispositivo NAT, una connessione VPN o AWS Connessione Direct Connect.

Le istanze presenti nel VPC non richiedono indirizzi IP pubblici per comunicare con gli endpoint API MemoryDB per Redis. Inoltre, le istanze non richiedono indirizzi IP pubblici per utilizzare le operazioni API MemoryDB disponibili. Il traffico di rete tra VPC e MemoryDB per Redis non esce dalla rete Amazon. Ogni endpoint dell'interfaccia è rappresentato da una o più interfacce di rete elastiche nelle sottoreti. Per maggiori informazioni sulle interfacce di rete elastiche, consulta le [interfacce di rete elastiche](#) nella Guida dell'utente di Amazon EC2.

- Per ulteriori informazioni sugli endpoint VPC, consulta [Endpoint VPC di interfaccia \(AWS PrivateLink\)](#) nella Amazon VPC User Guide.
- Per ulteriori informazioni sulle operazioni API MemoryDB, consulta [Operazioni API MemoryDB](#).

Dopo aver creato un endpoint VPC di interfaccia, se abiliti [DNS privato](#) hostnames per l'endpoint, l'endpoint MemoryDB predefinito (<https://memorydb.Region.amazonaws.com>) si risolve nell'endpoint VPC. Se non abiliti nomi host DNS privati, Amazon VPC fornisce un nome di endpoint DNS che puoi utilizzare nel formato seguente:

```
VPC_Endpoint_ID.memorydb.Region.vpce.amazonaws.com
```

Per ulteriori informazioni, consulta [Endpoint VPC di interfaccia \(AWS PrivateLink\)](#) nella Amazon VPC User Guide. MemoryDB supporta l'esecuzione di chiamate a tutte le sue [Operazioni API](#) all'interno del tuo VPC.

Note

I nomi host DNS privati possono essere abilitati per un solo endpoint VPC nel VPC. Se si desidera creare un endpoint VPC aggiuntivo, è necessario disabilitare il nome host DNS privato.

Considerazioni sugli endpoint VPC

Prima di impostare un endpoint VPC di interfaccia per endpoint MemoryDB per endpoint API Redis, verificare di esaminare [Proprietà e limitazioni degli endpoint dell'interfaccia](#) nella Amazon VPC User Guide. Tutte le operazioni API MemoryDB che sono rilevanti per la gestione delle risorse MemoryDB per Redis sono disponibili dal VPC utilizzando AWS PrivateLink. I criteri endpoint VPC sono supportati per endpoint MemoryDB API. Per impostazione predefinita, l'accesso completo alle operazioni API

MemoryDB è consentito attraverso l'endpoint. Per ulteriori informazioni, consultare [Controllo degli accessi ai servizi con endpoint VPC](#) in Guida per l'utente di Amazon VPC.

Creazione di un endpoint di interfaccia VPC per loAPI MemoryDB

È possibile creare un endpoint VPC per l'API MemoryDB per Redis utilizzando la console Amazon VPC o ilAWS CLI. Per ulteriori informazioni, consulta [Creazione di un endpoint di interfaccia](#) nella Guida per l'utente di Amazon VPC.

Una volta creato un endpoint VPC di interfaccia, è possibile abilitare nomi host DNS privati per l'endpoint. Quando lo fai, il MemoryDB predefinito per l'endpoint Redis (<https://memorydb.Region.amazonaws.com>) si risolve nell'endpoint VPC. Per ulteriori informazioni, consulta [Accesso a un servizio tramite un endpoint di interfaccia](#) in Guida per l'utente di Amazon VPC.

Creazione di una policy di endpoint VPC per loAPI Amazon MemoryDB

È possibile allegare una policy di endpoint all'endpoint VPC che controlla l'accesso all'API MemoryDB. La policy specifica quanto segue:

- Il principale che può eseguire operazioni.
- Le operazioni che possono essere eseguite.
- Le risorse sui cui si possono eseguire operazioni.

Per ulteriori informazioni, consulta [Controllo degli accessi ai servizi con endpoint VPC](#) in Guida per l'utente di Amazon VPC.

Example Policy di endpoint VPC per le operazioni API MemoryDB

Di seguito è riportato un esempio di una policy endpoint per l'API MemoryDB. Se collegato a un endpoint, questo criterio concede l'accesso alle operazioni API MemoryDB elencate per tutte le entità su tutte le risorse.

```
{
  "Statement": [{
    "Principal": "*",
    "Effect": "Allow",
    "Action": [
      "memorydb:CreateCluster",
      "memorydb:UpdateCluster",
```



```

    "memorydb:CreateSnapshot"
  ],
  "Resource": "*"
}]
}

```

Example Policy VPC Endpoint che nega tutti gli accessi da uno specificato AWS conto

La seguente policy di endpoint VPC nega all'account AWS **123456789012** l'accesso completo alle risorse utilizzando l'endpoint. La policy consente tutte le operazioni da altri account.

```

{
  "Statement": [{
    "Action": "*",
    "Effect": "Allow",
    "Resource": "*",
    "Principal": "*"
  },
  {
    "Action": "*",
    "Effect": "Deny",
    "Resource": "*",
    "Principal": {
      "AWS": [
        "123456789012"
      ]
    }
  }
]
}

```

Aggiornamenti del servizio in MemoryDB per Redis

MemoryDB for Redis monitora automaticamente il tuo parco istanze di cluster e nodi per applicare gli aggiornamenti del servizio non appena disponibili. In genere, imposti una finestra di manutenzione predefinita in modo che MemoryDB possa applicare questi aggiornamenti. Tuttavia, in alcuni casi questo approccio potrebbe risultare troppo rigido e vincolare per i flussi aziendali.

Con gli aggiornamenti del servizio, controlla quando e quali aggiornamenti vengono applicati. Puoi anche monitorare lo stato di avanzamento di questi aggiornamenti al cluster MemoryDB selezionato in tempo reale.

Gestire gli aggiornamenti del servizio

Gli aggiornamenti del servizio MemoryDB vengono rilasciati su base regolare. Se disponi di uno o più cluster idonei per tali aggiornamenti del servizio, riceverai notifiche tramite e-mail, SNS, eventi PHD (Personal Health Dashboard) e Amazon CloudWatch (Personal Health Dashboard) e Amazon CloudWatch. Gli aggiornamenti vengono inoltre mostrati sul [Pagina degli aggiornamenti del servizio](#) della console MemoryDB. Utilizzando questa dashboard, puoi visualizzare tutti gli aggiornamenti del servizio e il relativo stato per il parco istanze MemoryDB.

Puoi controllare quando applicare un aggiornamento prima dell'avvio di un aggiornamento automatico. Si consiglia di applicare gli aggiornamenti del tipo [aggiornamento della sicurezza](#) al più presto possibile, per garantire che MemoryDB sia sempre aggiornato con le patch di sicurezza correnti.

Le seguenti sezioni esplorano queste opzioni in dettaglio:

Argomenti

- [Applicazione degli aggiornamenti del servizio](#)

Applicazione degli aggiornamenti del servizio

Puoi iniziare ad applicare gli aggiornamenti del servizio al tuo parco istanze dal momento in cui gli aggiornamenti hanno `undisponiblestatus`. Gli aggiornamenti del servizio sono cumulativi. In altre parole, tutti gli aggiornamenti non ancora applicati sono inclusi nell'ultimo aggiornamento.

Se un aggiornamento del servizio è abilitato l'aggiornamento automatico, è possibile scegliere di non intraprendere alcuna azione quando diventa disponibile. MemoryDB pianificherà di applicare l'aggiornamento durante la finestra di manutenzione del cluster dopo il `Data` di inizio dell'aggiornamento automatico. Riceverai notifiche correlate per ogni fase dell'aggiornamento.

Note

È possibile applicare solo gli aggiornamenti del servizio che hanno `undisponibleopianificatastatus`.

Per ulteriori informazioni sulla revisione e sull'applicazione di eventuali aggiornamenti specifici del servizio ai cluster MemoryDB applicabili, consulta [Applicazione degli aggiornamenti del servizio tramite la console](#).

Quando è disponibile un nuovo aggiornamento del servizio per uno o più cluster MemoryDB, puoi utilizzare MemoryDB console, API o AWS CLI per applicare l'aggiornamento. Le sezioni seguenti illustrano le opzioni che puoi utilizzare per applicare gli aggiornamenti.

Applicazione degli aggiornamenti del servizio tramite la console

Per visualizzare l'elenco degli aggiornamenti del servizio, insieme ad altre informazioni, vai alla sezione **Aggiornamenti del servizio** Pagina della console.

1. Accedi alla **AWS Management Console** e apri la console MemoryDB per Redis all'indirizzo <https://console.aws.amazon.com/memorydb/>.
2. Nel riquadro di navigazione, scegli **Aggiornamenti del servizio**.

UNDER Dettagli dell'aggiornamento del servizio è possibile visualizzare quanto segue:

- **Nome dell'aggiornamento del servizio:** Il nome univoco dell'aggiornamento del servizio
- **Descrizione dell'aggiornamento:** Informazioni dettagliate sull'aggiornamento del servizio
- **Data di inizio dell'aggiornamento automatico:** Se questo attributo è impostato, MemoryDB inizierà a pianificare l'aggiornamento automatico dei cluster nelle finestre di manutenzione appropriate dopo tale data. Riceverai in anticipo le notifiche sull'esatta finestra di manutenzione programmata, che potrebbe non essere quella immediata dopo la **Data di inizio dell'aggiornamento automatico**. Puoi comunque applicare l'aggiornamento ai cluster ogni volta che preferisci. Se l'attributo non è impostato, l'aggiornamento del servizio non è abilitato l'aggiornamento automatico e MemoryDB non aggiornerà automaticamente i cluster.

Nella **Stato dell'aggiornamento del cluster** sezione, è possibile visualizzare un elenco di cluster in cui l'aggiornamento del servizio non è stato applicato o è stato appena applicato di recente. Per ogni cluster, puoi visualizzare quanto segue:

- **Cluster name (Nome del cluster):** Il nome del cluster
- **Nodi aggiornati:** Il rapporto tra i singoli nodi all'interno di un cluster specifico che sono stati aggiornati o rimangono disponibili per l'aggiornamento del servizio specifico.
- **Tipo di aggiornamento:** Il tipo di aggiornamento del servizio, che è uno dei **aggiornamento della sicurezza** o **Aggiornamento del motore**
- **Stato:** Lo stato dell'aggiornamento del servizio nel cluster, che è uno dei seguenti:
 - **disponibile:** L'aggiornamento è disponibile per il cluster richiesto.

- **in progress:** L'aggiornamento è in corso l'applicazione a questo cluster.
- **planned:** La data di aggiornamento è stata pianificata.
- **complete:** L'aggiornamento è stato applicato correttamente. Il cluster con uno stato completo verrà visualizzato per 7 giorni dopo il completamento.

Se hai scelto uno o tutti i cluster con lo stato pianificato, e poi hai scelto **Richiedi ora**, l'aggiornamento inizierà ad essere applicato su tali cluster.

Applicazione degli aggiornamenti del servizio tramite AWS CLI

Dopo aver ricevuto la notifica che gli aggiornamenti del servizio sono disponibili, puoi esaminarli e applicarli utilizzando la AWS CLI:

- Per recuperare una descrizione degli aggiornamenti del servizio disponibili, esegui il comando seguente:

```
aws memorydb describe-service-updates --status available
```

Per ulteriori informazioni, consulta [describe-service-updates](#).

- Per applicare un aggiornamento del servizio a un elenco di cluster, esegui il comando seguente:

```
aws memorydb batch-update-cluster --service-update  
ServiceUpdateNameToApply=sample-service-update --cluster-names cluster-1  
cluster2
```

Per ulteriori informazioni, consulta [batch update-cluster](#).

Riferimento

Gli argomenti presenti in questa sezione illustrano l'utilizzo dell'API MemoryDB e della sezione MemoryDB dell'AWS CLI. Sono inclusi anche i messaggi di errore e le notifiche di servizio più frequenti.

- [Utilizzo dell'API MemoryDB](#)
- [Informazioni di riferimento dell'API MemoryDB](#)
- [Sezione MemoryDB dell'AWS CLIRiferimento del](#)

Utilizzo dell'API MemoryDB

Questa sezione fornisce descrizioni delle attività per l'utilizzo e l'implementazione delle operazioni MemoryDB. Per una descrizione completa di queste operazioni, consulta [Informazioni di riferimento sull'API MemoryDB](#).

Argomenti

- [Uso dell'API query](#)
- [Librerie disponibili](#)
- [Risoluzione dei problemi delle applicazioni](#)

Uso dell'API query

Parametri di query

Le richieste basate su query HTTP sono richieste HTTP che utilizzano i verbi HTTP GET oppure POST e un parametro di query denominato `Action`.

Ogni richiesta di query deve includere alcuni parametri comuni per gestire l'autenticazione e la selezione di un'azione.

Alcune operazioni accettano elenchi di parametri. Questi elenchi sono specificati usando l'annotazione `param.n`. I valori di `n` sono numero a partire da 1.

Autenticazione delle richieste di query

È possibile inviare richieste di query solo tramite HTTPS ed è necessario includere una firma in ogni richiesta di query. Questa sezione descrive come creare la firma. Il metodo descritto nella seguente procedura è noto come Signature Version 4.

Di seguito sono riportate le fasi di base utilizzate per l'autenticazione delle richieste in AWS. Si presuppone che abbia eseguito la registrazione in AWS e disponga di un ID chiave di accesso e di una chiave di accesso segreta.

Processo di autenticazione delle query

1. Il mittente crea una richiesta in AWS.

2. Il mittente calcola la firma della richiesta, un hash con chiave per il codice di autenticazione di messaggi basati su hash (HMAC) con una funzione hash SHA-1, come definito nella successiva sezione di questo argomento.
3. Il mittente della richiesta invia ad AWS i dati della richiesta, la firma e l'ID chiave di accesso, ovvero l'identificatore per la chiave di accesso segreta in uso.
4. AWS utilizza l'ID chiave di accesso per cercare la chiave di accesso segreta.
5. AWS genera una firma a partire dai dati della richiesta e dalla chiave di accesso segreta utilizzando lo stesso algoritmo con cui viene calcolata la firma nella richiesta.
6. Se le firme corrispondono, la richiesta viene considerata autentica. Se il confronto non va a buon fine, la richiesta viene scartata e AWS invia una risposta di errore.

Note

Se una richiesta contiene un parametro `Timestamp`, la firma calcolata per la richiesta scade 15 minuti dopo il relativo valore.

Se una richiesta contiene un parametro `Expires`, la firma scade in corrispondenza dell'ora specificata dal parametro `Expires`.

Per calcolare la firma della richiesta

1. Creare la stringa di query in forma canonica necessaria successivamente durante questa procedura:
 - a. Ordinare i componenti della stringa di query UTF-8 per nome di parametro in base a un ordine naturale dei byte. I parametri possono avere fonte da GET URI o dal corpo POST (quando Content-Type è `application/x-www-form-urlencoded`).
 - b. Codificare in formato URL il nome e i valori di parametro in base alle seguenti regole:
 - i. Non codificare in formato URL i caratteri non riservati definiti da RFC 3986. I caratteri non riservati sono A-Z, a-z, 0-9, trattino (-), trattino basso (_), punto (.) e tilde (~).
 - ii. Codificare con codifica percentuale tutti gli altri caratteri con `%XY`, dove X e Y sono caratteri esadecimale (0-9 e A-F maiuscole).
 - iii. Codificare con codifica percentuale i caratteri UTF-8 estesi nel formato `%XY%ZA....`
 - iv. Codificare con codifica percentuale il carattere di spazio come `%20` (non utilizzare +, come negli schemi di codifica comuni).

- c. Separare i nomi di parametro codificati dai rispettivi valori codificati con il segno di uguale (=), ovvero il carattere ASCII 61, anche se il valore del parametro è vuoto.
 - d. Separare le coppie nome-valore con una E commerciale (&), codice ASCII 38.
2. Creare la stringa di firma in base al seguente esempio di grammatica ("\n" rappresenta una nuova riga ASCII).

```
StringToSign = HTTPVerb + "\n" +
ValueOfHostHeaderInLowercase + "\n" +
HTTPRequestURI + "\n" +
CanonicalizedQueryString <from the preceding step>
```

Il componente HTTPRequestURI è il componente del percorso assoluto HTTP dell'URI fino alla stringa di query (senza includerla). Se HTTPRequestURI è vuoto, utilizzare una barra (/).

3. Calcolare una firma HMAC compatibile con RFC 2104 utilizzando la stringa appena creata, la chiave di accesso segreta come chiave e SHA256 o SHA1 come algoritmo hash.

Per ulteriori informazioni, consulta <https://www.ietf.org/rfc/rfc2104.txt>.

4. Convertire il valore risultante in base64.
5. Includere il valore come valore del parametro Signature nella richiesta.

Di seguito è riportata una richiesta di esempio (le interruzioni di riga sono aggiunte per chiarezza).

```
https://memory-db.us-east-1.amazonaws.com/
?Action=DescribeClusters
&ClusterName=myCluster
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2021-01-01
```

Per la stringa di query precedente, è necessario calcolare la firma HMAC sulla seguente stringa.

```
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:
```

Il risultato è la richiesta firmata seguente.

```
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
```

Per informazioni dettagliate sul processo di firma e sul calcolo della firma della richiesta, consulta l'argomento [Processo di firma Signature Version 4](#) e i suoi sottoargomenti.

Librerie disponibili

AWS offre SDK (Software Development Kit) per sviluppatori software che preferiscono creare applicazioni utilizzando API specifiche del linguaggio anziché API di query. Questi SDK forniscono funzioni di base (non incluse nelle API), come autenticazione di richiesta, nuovi tentativi di richiesta e gestione degli errori per iniziare in modo più semplice. Sono disponibili SDK e risorse aggiuntive per i seguenti linguaggi di programmazione:

- [Java](#)
- [Windows e .NET](#)

- [PHP](#)
- [Python](#)
- [Ruby](#)

Per informazioni su altre lingue, consulta [Documentazione di esempio di codice e librerie](#).

Risoluzione dei problemi delle applicazioni

MemoryDB fornisce errori specifici e descrittivi per aiutarti a risolvere i problemi mentre interagisci con l'API MemoryDB.

Errore durante il recupero

In genere, si desidera che l'applicazione verifichi se una richiesta ha generato un errore prima di trascorrere del tempo a elaborare i risultati. Il modo più semplice per determinare se si è verificato un errore consiste nel cercare un `Error` nella risposta dall'API MemoryDB.

La sintassi XPath fornisce un modo semplice per cercare la presenza di un `Error` nodo, così come un modo semplice per recuperare il codice di errore e il messaggio. Il seguente snippet di codice usa Perl e il modulo `XML::XPath` per determinare se si è verificato un errore durante una richiesta. Se si è verificato un errore, il codice stampa il primo codice di errore e il messaggio nella risposta.

```
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"; }
```

Suggerimenti per la risoluzione dei problemi

Si consigliano i seguenti processi per diagnosticare e risolvere i problemi con l'API MemoryDB.

- Verificare che MemoryDB sia eseguito correttamente.

A tale scopo, apri una finestra del browser e invia una richiesta di query al servizio MemoryDB, ad esempio `https://memory-db.us-east-1.amazonaws.com`. Un'eccezione `MissingAuthenticationTokenException` o `UnknownOperationException` conferma che il servizio è disponibile e risponde alle richieste.

- Verificare la struttura della richiesta.

Ogni operazione MemoryDB ha una pagina di riferimento nella documentazione di riferimento nella documentazione Informazioni di riferimento sull'API MemoryDB. Controllare nuovamente che si stia usando i parametri correttamente. Per avere delle idee su cosa potrebbe essere sbagliato, guarda le richieste di esempio o gli scenari utente per vedere se quegli esempi stanno eseguendo operazioni simili.

- Controllare il forum.

MemoryDB ha un forum di discussione in cui è possibile cercare soluzioni ai problemi che altri hanno trovato lungo il percorso. Per visualizzare il forum, accedi all'indirizzo

<https://forums.aws.amazon.com/>.

Quote per Amazon MemoryDB per Redis

Il tuo AWS account ha delle quote predefinite, precedentemente denominate limiti, per ogni servizio. AWS Salvo diversa indicazione, ogni quota si applica a una regione specifica. Se per alcune quote è possibile richiedere aumenti, altre quote non possono essere modificate.

Per richiedere un aumento delle quote, consultare [Richiesta di aumento delle quote](#) nella Guida per l'utente di Service Quotas. Se la quota non è ancora disponibile in Service Quotas, utilizza il [modulo di incremento dei limiti](#).

Il tuo AWS account ha le seguenti quote relative a MemoryDB.

Risorsa	Di default
Nodi per regione	300
Nodi per cluster per tipo di istanza	90
Nodi per partizione	6
Gruppi di parametri per regione	150
Gruppi di sottoreti per regione	150
Sottoreti per gruppo di sottoreti	20
Utenti per gruppo di utenti	100
Numero totale di utenti	1000
Numero di gruppi di utenti	100

Cronologia dei documenti per la Guida per l'utente di MemoryDB

Nella tabella seguente vengono descritte le varie versioni della documentazione per MemoryDB.

Modifica	Descrizione	Data
MemoryDB supporta ora l'autenticazione degli utenti tramite IAM	Autenticazione IAM consente di autenticare una connessione a MemoryDB per Redis utilizzando identità. AWS Identity and Access Management. Ciò consente di consolidare il modello di sicurezza e semplificare molte attività di sicurezza amministrative. Per ulteriori informazioni, consulta Autenticazione con IAM .	10 maggio 2023
MemoryDB supporta Redis 7	Questa versione introduce diverse nuove funzionalità in MemoryDB per Redis: funzioni Redis, miglioramenti ACL, Pub/Sub partizionato e multiplexing I/O migliorato. Per ulteriori informazioni, consulta Versioni del motore Redis .	9 maggio 2023
MemoryDB ora offre nodi riservati	I nodi riserva offrono una notevole discount prezzo rispetto ai nodi on demand. I nodi riserva non sono nodi fisica, ma piuttosto si tratta di uno discount sulla fattura applicato all'uso dei nodi on	27 dicembre 2022

demand nell'account. Per ulteriori informazioni, consulta nodi memoria [riservata di MemoryDB](#).

[MemoryDB ora supporta Data Tiering](#)

MemoryDB for Redis. È possibile utilizzare il tiering di dati come metodo a costo contenuto per dimensionare i cluster fino a centinaia di terabyte di capacità. Per ulteriori informazioni, consulta [Tiering di dati](#).

3 novembre 2022

[MemoryDB supporta il formato JSON \(JavaScript Object Notation\) nativo](#)

Il formato JSON (JavaScript Object Notation) nativo è un modo semplice e senza schemi per codificare set di dati complessi nei cluster Redis. Puoi archiviare e accedere in modo nativo ai dati utilizzando il formato JavaScript Object Notation (JSON) nei cluster Redis e aggiornare i dati JSON memorizzati in tali cluster senza la necessità di gestire codice personalizzato per serializzarlo e deserializzarlo e deserializzarlo e deserializzarlo. Per ulteriori informazioni, consulta [Nozioni di base su JSON](#).

25 maggio 2022

[MemoryDB ora supporta AWS PrivateLink](#)

AWS PrivateLink consente di accedere privatamente alle operazioni dell'API MemoryDB senza un gateway Internet, un dispositivo NAT, una connessione VPN o AWS una connessione Direct Connect. Per ulteriori informazioni, consulta [API MemoryDB ed endpoint VPC di interfaccia \(\)](#).
AWS PrivateLink

24 gennaio 2022

[Versione iniziale](#)

Versione iniziale della Guida per l'utente di MemoryDB User. Per ulteriori informazioni, consulta la pagina [Che cos'è MemoryDB?](#)

19 agosto 2021

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.