



Guida per l'utente

Amazon ElastiCache per Redis



Versione API 2015-02-02

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

Amazon ElastiCache per Redis: Guida per l'utente

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

I marchi e il trade dress di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in qualsiasi modo che possa causare confusione tra i clienti o in qualsiasi modo 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

Che cos'è ElastiCache per Redis?	1
Cache serverless	1
Cluster progettati autonomamente	2
Servizi correlati	2
Come funziona	3
Motori di cache e caching	3
Scelta dell'implementazione	8
Confronto delle funzionalità	10
Risorse ElastiCache	15
AWSRegioni e zone di disponibilità	17
Casi d'uso	18
Datastore in memoria	19
Classifiche per videogiochi (set ordinati Redis)	20
Messaggistica (pub/sub Redis)	22
Data consigliati (hash redis)	25
Altri usi di Redis	25
Testimonianze dei clienti su ElastiCache	25
Guida introduttiva a ElastiCache Redis	26
Configurazione	26
Registrati per un Account AWS	26
Creazione di un utente amministratore	27
Concessione dell'accesso programmatico	28
Impostazione delle autorizzazioni	30
Configurazione di EC2	31
Assegnazione dell'accesso di rete	31
Configurazione di redis-cli	32
Crea una cache.	33
Lettura e scrittura di dati	34
Eliminazione	36
Fasi successive	37
Endamenti su ElastiCache e AWS SDK	37
Python e ElastiCache	37
Tutorial: Configurazione di una funzione Lambda per accedere ad Amazon in un ElastiCache Amazon VPC	55

Fase 1: Creare una cache serverless	55
Passaggio 2: creazione di una funzione Lambda	58
Fase 3: esecuzione del test della funzione Lambda	62
Fase 4: Pulizia (opzionale)	63
Progettazione del proprio cluster ElastiCache	65
Componenti e caratteristiche	65
Nodi	66
ElastiCache per gli shard Redis	66
ElastiCache per i cluster Redis	67
ElastiCache per la replica Redis	69
AWS Regioni e zone di disponibilità	71
ElastiCache per gli endpoint Redis	72
Gruppi di parametri	72
ElastiCache per la sicurezza Redis	73
Gruppi di sottoreti	73
ElastiCache per i backup Redis	74
Eventi	74
Terminologia di ElastiCache per Redis	75
Progettazione del proprio cluster	78
Configurazione	78
Fase 1: creare un gruppo di sottoreti	78
Fase 2: creazione di un cluster	81
Fase 3: autorizzazione dell'accesso al cluster	88
Fase 4: connessione a un nodo del cluster	91
Passaggio 5: Eliminazione di un cluster	99
Tutorial e video	100
A questo punto come si può procedere?	106
Gestione di nodi	107
Visualizzazione dello stato ElastiCache del nodo	108
Nodi e partizioni Redis	114
Connessione ai nodi	116
Tipi di nodi supportati	119
Riavvio dei nodi (solo modalità cluster disabilitata)	129
Sostituzione dei nodi	131
Nodi riservati	138
Migrazione dei nodi della generazione precedente	150

Gestione dei cluster	153
Scelta del tipo di rete	155
Tiering di dati	159
Preparazione di un cluster	166
Creazione di un cluster	173
Visualizzazione dei dettagli di un cluster	183
Modifica di un cluster	195
Aggiunta di nodi a un cluster	200
Rimozione di nodi da un cluster	208
Annullamento delle operazioni in attesa di aggiunta e di eliminazione dei nodi	216
Eliminazione di un cluster	217
Accesso al cluster o gruppo di replica	220
Individuazione degli endpoint di connessione	226
Partizioni	238
Confronto tra cache progettate autonomamente Memcached e Redis	243
Migrazione ElastiCache	248
Panoramica	249
Fasi della migrazione	249
Preparazione dei nodi di fonte e di destinazione per la migrazione	250
Test della migrazione dei dati	251
Avvio della migrazione	252
Verifica dell'avanzamento della migrazione dei dati	253
Completamento della migrazione dei dati	254
Esecuzione della migrazione dei dati online utilizzando la console	255
Scelta di regioni e zone di disponibilità	256
Ubicazione dei nodi	258
Regioni ed endpoint supportati	258
Utilizzo di Local Zones	263
Utilizzo Outposts	265
Lavorare con ElastiCache	269
Snapshot e ripristino	269
Vincoli	270
Impatto sulle prestazioni dei backup di cluster progettati autonomamente	271
Pianificazione di backup automatici	272
Esecuzione di backup manuali	273
Creazione di un backup finale	279

Descrizione dei backup	282
Copia di backup	284
Esportazione di un backup	286
Ripristino da un backup	294
Eliminazione di un backup	296
Assegnazione di tag di backup	297
Inizializzazione di un cluster progettato autonomamente con un backup	299
Versioni e aggiornamenti del motore	308
Versioni e aggiornamenti del motore	309
Versioni di Redis supportate	314
Pianificazione della fine del ciclo di vita delle versioni di Redis	328
Come aggiornare la versione di un motore	312
Come risolvere il blocco di un aggiornamento del motore	312
Differenze di compatibilità e comportamento delle versioni principali	331
Best practice e strategie di caching	335
Utilizzo di Redis	335
Best practice con i client Redis	375
Best practice per l'utilizzo di cluster progettati autonomamente	402
Best practice per Redis	415
Strategie di caching	417
Gestione del cluster progettato autonomamente	422
Auto Scaling ElastiCache per cluster Redis	423
Modifica della modalità cluster	471
Replica tra regioni AWS mediante Global Datastore	474
Alta disponibilità utilizzando gruppi di replica	501
Gestione della manutenzione	589
Configurazione dei parametri di motore con i gruppi di parametri	592
Scalabilità per Redis ElastiCache	693
Scalabilità ElastiCache senza server	693
Impostazione dei limiti di dimensionamento per gestire i costi	694
Prescalabilità con Serverless ElastiCache	694
Impostazione dei limiti di scalabilità utilizzando la console e AWS CLI	695
Scalabilità per cluster progettati autonomamente da Redis ElastiCache	697
Nozioni di base di JSON in ElastiCache per Redis	770
Panoramica del tipo di dati JSON di Redis	771
Comandi JSON	783

Assegnazione di tag delle risorse ElastiCache	825
Monitoraggio dei costi con i tag	836
Gestione dei tag con la AWS CLI	838
Gestione dei tag con l'API di ElastiCache	841
Amazon ElastiCache Well-Architected Lens	844
Pilastro dell'eccellenza operativa	845
Pilastro della sicurezza	853
Pilastro dell'affidabilità	860
Pilastro dell'efficienza delle prestazioni	866
Pilastro dell'ottimizzazione dei costi	877
Risoluzione dei problemi	883
Problemi di connessione	883
Errori del client Redis	884
Risoluzione dei problemi di latenza elevata in Serverless ElastiCache	885
Risoluzione dei problemi di limitazione in Serverless ElastiCache	886
Argomenti correlati	887
Ulteriori passaggi per la risoluzione dei problemi	887
Gruppi di sicurezza	888
Liste di controllo accessi (ACL) di rete	888
Tabelle di instradamento	890
Risoluzione DNS	890
Identificazione dei problemi con la diagnostica lato server	891
Convalida della connettività di rete	896
Limiti relativi alla rete	899
Utilizzo CPU	900
Connessioni terminate dal lato server	903
Risoluzione dei problemi lato client per istanze Amazon EC2	905
Dissezione del tempo necessario per completare una singola richiesta	906
Sicurezza	909
Protezione dei dati	910
Sicurezza dei dati in Amazon ElastiCache	910
Riservatezza del traffico Internet	982
Amazon VPC e sicurezza di ElastiCache	983
API Amazon ElastiCache ed endpoint VPC di interfaccia (AWS PrivateLink)	1006
Sottoreti e gruppi di sottoreti	1009
Identity and Access Management	1017

Destinatari	1017
Autenticazione con identità	1018
Gestione dell'accesso con policy	1022
Funzionamento di Amazon ElastiCache con IAM	1024
Esempi di policy basate su identità	1031
Risoluzione dei problemi	1035
Controllo accessi	1037
Panoramica sulla gestione degli accessi	1038
Convalida della conformità	1081
Ulteriori informazioni	1083
Resilienza	1083
Limitazione dell'impatto degli errori	1083
Sicurezza dell'infrastruttura	1087
Aggiornamenti di servizio	1087
Gestione degli aggiornamenti del servizio	1088
Vulnerabilità di sicurezza risolte	1093
Registrazione di log e monitoraggio	1096
Metriche ed eventi serverless	1096
Metriche serverless	1096
Eventi serverless	1105
Metriche ed eventi di cluster progettati autonomamente	1120
Metriche dei cluster progettati autonomamente	1120
Eventi di cluster progettati autonomamente	1120
Consegna dei registri	1129
Monitoraggio dell'utilizzo	1142
Monitoraggio degli eventi di Amazon SNS	1171
Registrazione delle chiamate all'API Amazon ElastiCache con AWS CloudTrail	1188
Informazioni su Amazon ElastiCache in CloudTrail	1188
Informazioni sulle voci dei file di log di Amazon ElastiCache	1189
Quote	1194
Riferimento	1196
Utilizzo dell'API ElastiCache	1196
Uso dell'API query	1196
Librerie disponibili	1200
Risoluzione dei problemi delle applicazioni	1200
Impostazione dell'interfaccia AWS CLI ElastiCache	1201

Prerequisiti	1202
Ottenere gli strumenti a riga di comando	1203
Configurazione degli strumenti	1204
Fornire le credenziali per gli strumenti	1205
Variabili di ambiente	1206
Messaggi di errore	1207
Notifiche	1209
Notifiche generali ElastiCache	1209
ElastiCache per notifiche specifiche Redis	1209
ElastiCache per la cronologia della documentazione Redis	1210
AWS Glossario	1244
.....	mccxlv

Cos'è Amazon ElastiCache per Redis?

Benvenuto nella Guida ElastiCache per l'utente di Amazon for Redis. Amazon ElastiCache è un servizio Web che semplifica la configurazione, la gestione e la scalabilità di un archivio dati in memoria distribuito o di un ambiente di cache nel cloud. Fornisce una soluzione di caching scalabile ad alte prestazioni e a costi contenuti. Allo stesso tempo, aiuta a rimuovere la complessità associata all'implementazione e alla gestione di un ambiente di cache distribuita.

Puoi utilizzare Amazon ElastiCache in due formati. Puoi iniziare con una cache serverless o scegliere di progettare il tuo cluster di cache.

Note

Amazon ElastiCache funziona con entrambi i motori Redis e Memcached. Utilizza la guida relativa al motore che ti interessa. Se non sei certo di quale motore utilizzare, consulta [Confronto tra cache progettate autonomamente Memcached e Redis](#) in questa guida.

Cache serverless

ElastiCache for Redis offre un caching senza server, che semplifica l'aggiunta e il funzionamento di una cache basata su Redis per l'applicazione. ElastiCache for Redis Serverless consente di creare una cache ad alta disponibilità in meno di un minuto ed elimina la necessità di effettuare il provisioning di istanze o configurare nodi o cluster. Gli sviluppatori possono creare una cache serverless specificando il nome della cache utilizzando la ElastiCache console, l'SDK o la CLI.

ElastiCache for Redis Serverless elimina inoltre la necessità di pianificare e gestire la capacità di caching. ElastiCache for Redis monitora costantemente la memoria della cache, l'elaborazione e la larghezza di banda di rete utilizzata dall'applicazione e si adatta alle esigenze dell'applicazione. ElastiCache for Redis offre agli sviluppatori un'esperienza endpoint semplice, astruendo l'infrastruttura di cache sottostante e la progettazione del cluster. ElastiCache for Redis gestisce il provisioning dell'hardware, il monitoraggio, la sostituzione dei nodi e l'applicazione di patch software in modo automatico e trasparente, in modo che tu possa concentrarti sullo sviluppo delle applicazioni anziché sull'utilizzo della cache.

ElastiCache for Redis Serverless è compatibile con Redis 7.1 e versioni successive.

Progettazione del proprio cluster per Redis ElastiCache

Se hai bisogno di un controllo granulare sul tuo cluster ElastiCache per Redis, puoi scegliere di progettare il tuo cluster Redis con. ElastiCache consente di progettare il cluster, scegliendo il tipo di nodo, il numero di nodi e il posizionamento dei nodi nelle zone di disponibilità del cluster. AWS Trattandosi ElastiCache di un servizio completamente gestito, gestisce automaticamente il provisioning dell'hardware, il monitoraggio, la sostituzione dei nodi e l'applicazione di patch software per il cluster.

La progettazione personalizzata ElastiCache per il cluster Redis offre maggiore flessibilità e controllo sui cluster. Ad esempio, puoi scegliere di gestire un cluster con disponibilità per singola zona di disponibilità o multi-AZ a seconda delle tue esigenze. Puoi anche scegliere di eseguire Redis in modalità cluster, abilitando il dimensionamento orizzontale oppure senza la modalità cluster per il dimensionamento verticale. Quando si progettano i propri cluster, è responsabilità dell'utente scegliere correttamente il tipo e il numero di nodi per garantire che la cache disponga della capacità sufficiente a soddisfare le esigenze dell'applicazione. È inoltre possibile scegliere quando applicare le nuove patch del software al cluster Redis.

Quando si progetta un cluster personalizzato ElastiCache per Redis, è possibile scegliere di eseguire Redis 3.0 e versioni successive.

Servizi correlati

[Amazon MemoryDB per Redis](#)

Per decidere se utilizzare ElastiCache per Redis o Amazon MemoryDB per Redis, considerare i seguenti confronti:

- ElastiCache per Redis è un servizio comunemente utilizzato per memorizzare nella cache i dati di altri database e datastore utilizzando Redis. È opportuno considerare ElastiCache per Redis per memorizzare nella cache carichi di lavoro in cui si desidera accelerare l'accesso ai dati con il database primario o il datastore esistente (prestazioni di lettura e scrittura in microsecondi). Occorre anche considerare ElastiCache per Redis per i casi d'uso in cui si desidera utilizzare le strutture dati e le API Redis per accedere ai dati archiviati in un database o datastore primario.
- Amazon MemoryDB per Redis è un database in memoria durevole 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.

[Amazon RDS](#)

ElastiCache per Redis può aiutarti a contenere i costi del database archiviando i dati a cui accedi di frequente in una cache. Se la tua applicazione richiede un'alta velocità di trasmissione effettiva in lettura, puoi ottenere dimensionamento elevato, prestazioni veloci e costi di archiviazione di dati ridotti utilizzando ElastiCache, invece di dimensionare il database sottostante.

Come funziona

Qui puoi trovare una panoramica dei componenti principali di una distribuzione ElastiCache per Redis.

Motori di cache e caching

Una cache è un archivio dati in memoria che è possibile utilizzare per archiviare i dati memorizzati nella cache. In genere, l'applicazione memorizza nella cache i dati a cui si accede di frequente per ottimizzare i tempi di risposta. ElastiCache for Redis offre due opzioni di implementazione: cluster serverless e cluster progettati autonomamente. Per informazioni, consultare [Scelta tra le opzioni di implementazione](#).

Note

Amazon ElastiCache funziona con entrambi i motori Redis e Memcached. Utilizza la guida relativa al motore che ti interessa. Se non sei certo di quale motore utilizzare, consulta [Confronto tra cache progettate autonomamente Memcached e Redis](#) in questa guida.

Argomenti

- [Come funziona ElastiCache per Redis](#)
- [Prezzi delle dimensioni](#)

- [ElastiCache per i backup Redis](#)

Come funziona ElastiCache per Redis

ElastiCache per Redis Serverless

ElastiCache per Redis Serverless consente di creare una cache senza preoccuparsi della pianificazione della capacità, della gestione dell'hardware o della progettazione del cluster. È sufficiente fornire il nome della cache per ricevere un singolo endpoint che puoi configurare nel client Redis per iniziare ad accedere alla cache.

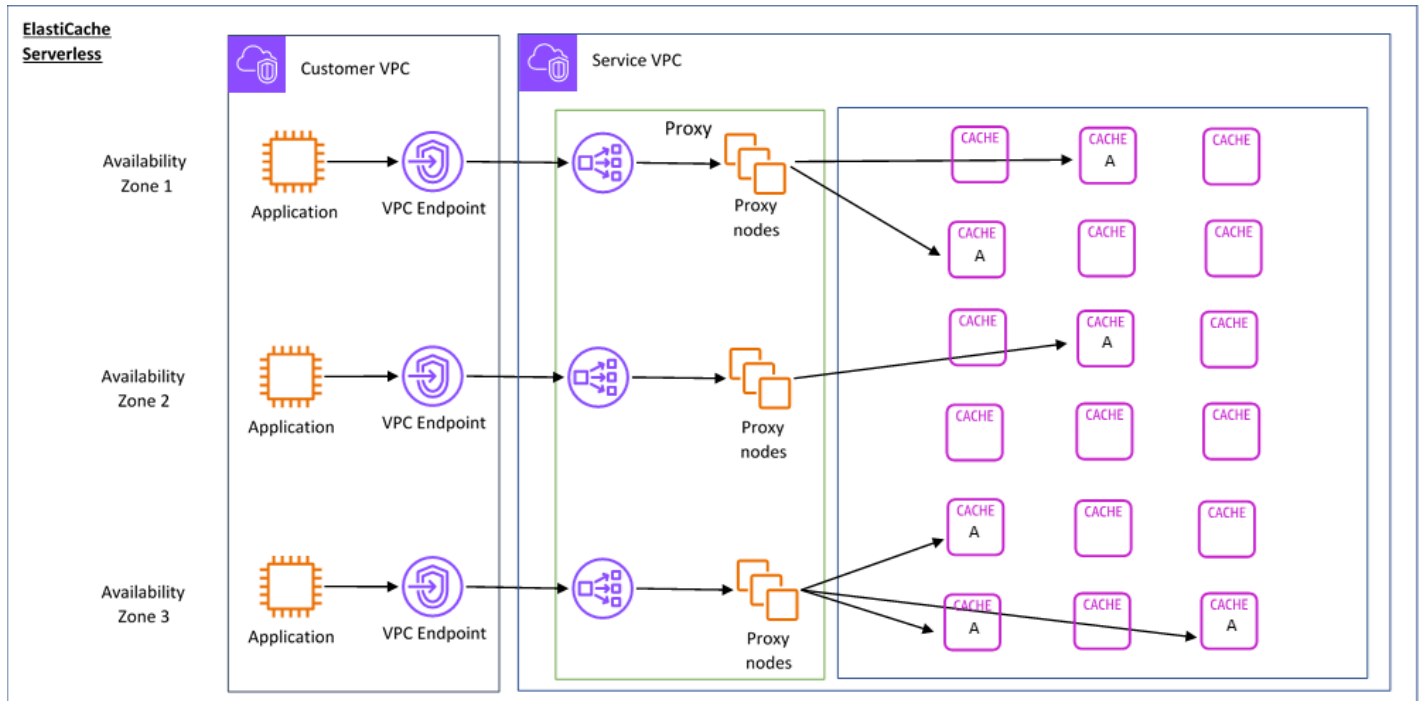
Note

ElastiCache for Redis Serverless esegue Redis in modalità cluster ed è compatibile solo con i client Redis che supportano sia TLS che il protocollo cluster Redis.

Vantaggi principali

- Nessuna pianificazione della capacità: ElastiCache Serverless elimina la necessità di pianificare la capacità. ElastiCache Serverless monitora continuamente la memoria, l'elaborazione e l'utilizzo della larghezza di banda di rete della cache e scala sia verticalmente che orizzontalmente. Consente di aumentare le dimensioni di un nodo di cache, avviando parallelamente un'operazione di scalabilità orizzontale per garantire che la cache possa dimensionarsi per soddisfare i requisiti dell'applicazione in ogni momento.
- Ray-per-use: Con ElastiCache Serverless, paghi per i dati archiviati e di calcolo utilizzati dal carico di lavoro sulla cache. Per informazioni, consulta [Prezzi delle dimensioni](#).
- Alta disponibilità: ElastiCache Serverless replica automaticamente i dati su più zone di disponibilità (AZ) per un'elevata disponibilità. Monitora automaticamente i nodi di cache sottostanti e li sostituisce in caso di errore. Offre uno SLA con disponibilità del 99,99% per ogni cache.
- Aggiornamenti automatici del software: ElastiCache Serverless aggiorna automaticamente la cache all'ultima versione del software secondario e all'ultima versione del software con patch senza alcun impatto sulla disponibilità dell'applicazione. Quando sarà disponibile una nuova versione principale di Redis, ElastiCache ti invierà una notifica.
- Sicurezza: serverless esegue sempre la crittografia dei dati in transito e a riposo. È possibile utilizzare una chiave gestita dal servizio o la chiave gestita dal cliente per crittografare i dati a riposo.

Il diagramma seguente illustra come funziona ElastiCache Serverless.



Quando crei una nuova cache serverless, ElastiCache crea un endpoint Virtual Private Cloud (VPC) nelle sottoreti di tua scelta nel tuo VPC. L'applicazione può connettersi alla cache tramite questi endpoint VPC.

Con ElastiCache Serverless ricevi un singolo endpoint DNS a cui si connette l'applicazione. Quando richiedi una nuova connessione all'endpoint, ElastiCache Serverless gestisce tutte le connessioni alla cache tramite un livello proxy. Il livello del proxy aiuta a ridurre la complessità della configurazione del client perché il client non ha bisogno di individuare nuovamente la topologia del cluster in caso di modifiche al cluster sottostante. Il livello del proxy è un set di nodi proxy che gestiscono le connessioni utilizzando un Network Load Balancer. Quando l'applicazione crea una nuova connessione alla cache, la richiesta viene inviata a un nodo proxy dal Network Load Balancer. Quando l'applicazione esegue i comandi della cache, il nodo proxy connesso all'applicazione elabora le richieste su un nodo di cache. Il livello del proxy estrae la topologia e i nodi del cluster di cache dal client. Ciò consente di ElastiCache bilanciare in modo intelligente il carico, scalare e aggiungere nuovi nodi di cache, sostituire i nodi di cache in caso di guasto e aggiornare il software sui nodi di cache, il tutto senza influire sulla disponibilità sull'applicazione o dover reimpostare le connessioni.

Cluster progettati autonomamente ElastiCache

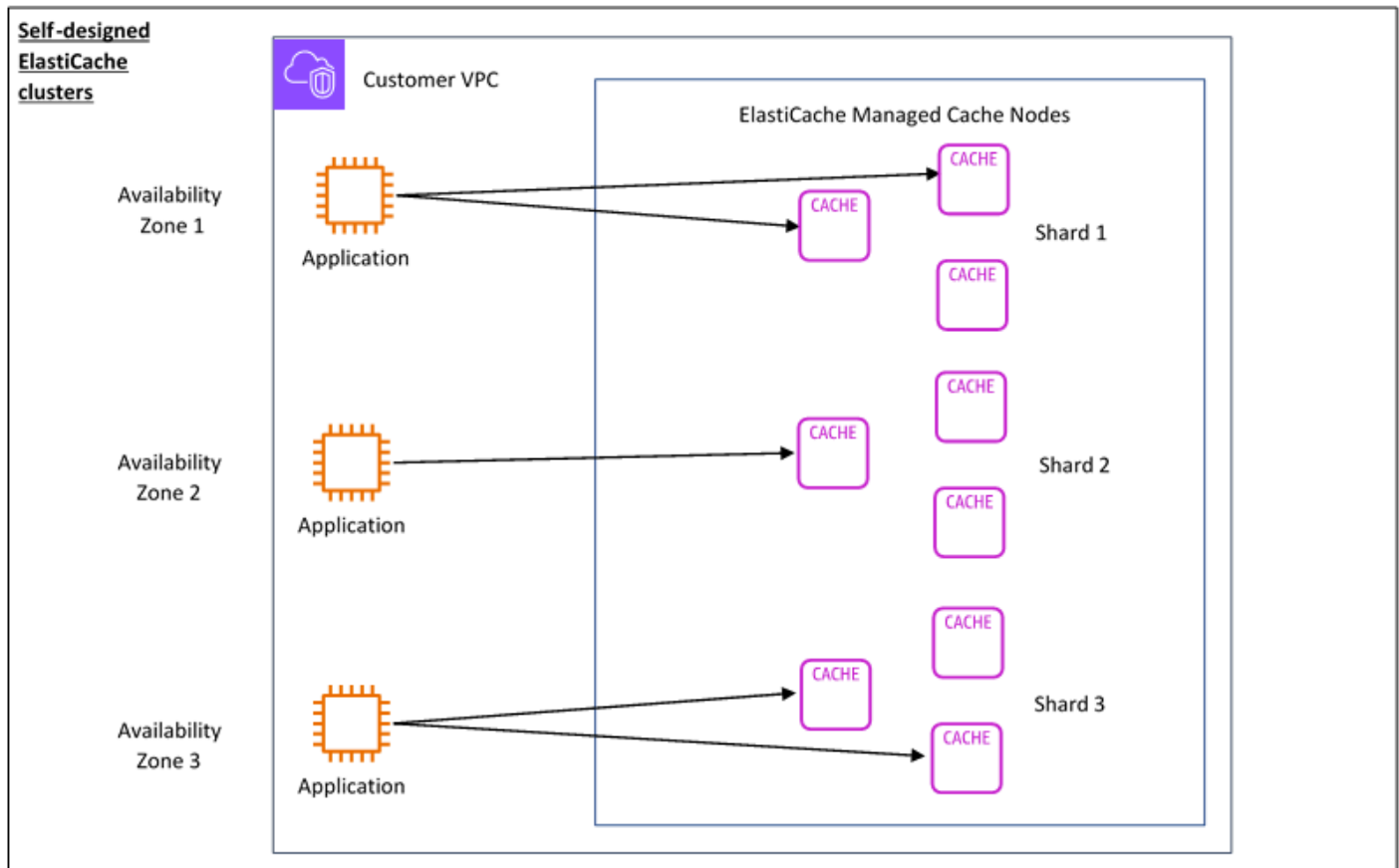
Puoi scegliere di progettare i tuoi ElastiCache cluster scegliendo una famiglia di nodi di cache, le dimensioni e il numero di nodi per il cluster. La progettazione del proprio cluster offre un controllo più

granulare e consente di scegliere il numero di partizioni nella cache e il numero di nodi (primari e di replica) in ogni partizione. Puoi scegliere di utilizzare Redis in modalità cluster creando un cluster con più partizioni oppure in modalità non cluster con una singola partizione.

Vantaggi principali

- **Progetta il tuo cluster:** con ElastiCache, puoi progettare il tuo cluster e scegliere dove posizionare i nodi di cache. Ad esempio, se un'applicazione compromette l'alta disponibilità con una bassa latenza, puoi scegliere di implementare i nodi di cache in un'unica zona di disponibilità. In alternativa, puoi progettare il tuo cluster con nodi su più zone di disponibilità per ottenere l'alta disponibilità.
- **Controllo granulare:** quando progetti il tuo cluster, hai un maggiore controllo sull'ottimizzazione delle impostazioni della cache. Ad esempio, puoi utilizzare [Parametri specifici di Redis](#) per configurare il motore di cache.
- **Dimensionamento verticale e orizzontale:** puoi scegliere di dimensionare manualmente il cluster aumentando o diminuendo la dimensione del nodo di cache quando necessario. Puoi anche dimensionare orizzontalmente aggiungendo nuove partizioni o altre repliche alle partizioni. Puoi anche utilizzare la funzionalità Auto-Scaling per configurare il ridimensionamento in base a una pianificazione o il ridimensionamento in base a metriche come l'utilizzo della CPU e della memoria nella cache.

Il diagramma seguente illustra come funzionano i cluster progettati autonomamente. ElastiCache



Prezzi delle dimensioni

È possibile eseguire la distribuzione in due opzioni. ElastiCache Serverless, si paga in base all'utilizzo dei dati archiviati in GB all'ora e per l'elaborazione in unità di elaborazione (ECPU). ElastiCache Quando scegli di progettare uno personalizzato ElastiCache per i cluster Redis, paghi in base all'ora di utilizzo del nodo di cache. Per i dettagli dei prezzi, consulta [questa pagina](#).

Archiviazione di dati

Paghi per i dati archiviati in ElastiCache Serverless e fatturati in gigabyte/ora (GB-h). ElastiCache Serverless monitora continuamente i dati archiviati nella cache, campionandoli più volte al minuto, e calcola una media oraria per determinare l'utilizzo dell'archiviazione dei dati della cache in GB-h. Ogni cache ElastiCache Serverless viene misurata per un minimo di 1 GB di dati archiviati.

ElastiCache Unità di elaborazione (ECPU)

Paghi per le richieste Redis che la tua applicazione esegue su ElastiCache Serverless in ElastiCache Processing Units (ECPU), un'unità che include sia il tempo vCPU che i dati trasferiti.

- Le operazioni di lettura e scrittura semplici richiedono 1 ECPU per ogni kilobyte (KB) di dati trasferiti. Ad esempio, un comando GET che trasferisce fino a 1 KB di dati consuma 1 ECPU. Una richiesta SET che trasferisce 3,2 KB di dati consuma 3,2 ECPU.
- I comandi che richiedono un tempo di vCPU aggiuntivo consumano proporzionalmente più ECPU. Ad esempio, se l'applicazione utilizza il [comando HMGET](#) di Redis e consuma 3 volte il tempo di vCPU come comando SET/GET, consuma 3 ECPU.
- I comandi che consumano più tempo di vCPU e trasferiscono più dati consumano le ECPU in base alla dimensione più alta tra le due. Ad esempio, se l'applicazione utilizza il comando HMGET, consuma 3 volte il tempo di vCPU di un comando SET/GET e trasferisce 3,2 KB di dati, consuma 3,2 ECPU. In alternativa, se trasferisce solo 2 KB di dati, consuma 3 ECPU.

ElastiCache Serverless emette una nuova metrica chiamata `ElastiCacheProcessingUnits` che ti aiuta a comprendere le ECPU utilizzate dal tuo carico di lavoro.

Ore per nodo

Puoi scegliere di progettare il tuo cluster di cache Redis scegliendo la famiglia di nodi EC2, la dimensione, il numero di nodi e il posizionamento nelle zone di disponibilità. Quando si progetta autonomamente il cluster, si paga per ora di utilizzo per ogni nodo di cache.

ElastiCache per i backup Redis

Un backup è una point-in-time copia di una cache Redis. ElastiCache consente di eseguire un backup dei dati in qualsiasi momento o di configurare backup automatici. I backup possono essere utilizzati per ripristinare una cache esistente o per inizializzarne una nuova. I backup sono costituiti da tutti i dati di una cache e da alcuni metadati. Per ulteriori informazioni, consulta [. Snapshot e ripristino.](#)

Scelta tra le opzioni di implementazione

Amazon ElastiCache offre due opzioni di distribuzione:

- Cache serverless
- Cluster progettati autonomamente

Per un elenco dei comandi supportati per entrambi, consulta [Comandi Redis supportati e limitati.](#)

Cache serverless

Amazon ElastiCache Serverless semplifica la creazione di cache e si ridimensiona istantaneamente per supportare le applicazioni più esigenti dei clienti. Con ElastiCache Serverless, puoi creare una cache altamente disponibile e scalabile in meno di un minuto, eliminando la necessità di fornire, pianificare e gestire la capacità del cluster di cache. ElastiCache Serverless archivia automaticamente i dati in modo ridondante su tre zone di disponibilità e fornisce un Service Level Agreement (SLA) di disponibilità del 99,99%. I backup sono intercompatibili e possono essere esportati e ripristinati da cluster progettati autonomamente.

Cluster progettati autonomamente

Se hai bisogno di un controllo granulare sul tuo cluster ElastiCache for Redis, puoi scegliere di progettare il tuo cluster Redis con ElastiCache. ElastiCache consente di gestire un cluster basato su nodi, scegliendo il tipo di nodo, il numero di nodi e il posizionamento dei nodi tra le zone di disponibilità del cluster. AWS Trattandosi ElastiCache di un servizio completamente gestito, aiuta a gestire il provisioning dell'hardware, il monitoraggio, la sostituzione dei nodi e l'applicazione di patch software per il cluster. I cluster progettati autonomamente possono essere progettati per fornire uno SLA di disponibilità fino al 99,99%. I backup sono compatibili tra loro e possono essere esportati e ripristinati dalle cache Serverless.

Scelta tra le opzioni di implementazione

Scegli la cache serverless se:

- Stai creando una cache per carichi di lavoro nuovi o difficili da prevedere.
- Si dispone di un traffico delle applicazioni non prevedibile.
- desideri iniziare a usare una cache nel modo più semplice.

Scegli di progettare il tuo ElastiCache cluster se:

- Stai già utilizzando ElastiCache Serverless e desideri un controllo più preciso sul tipo di nodo su cui è in esecuzione Redis, sul numero di nodi e sul posizionamento dei nodi.
- Ti aspetti che il traffico delle tue applicazioni sia relativamente prevedibile e desideri un controllo granulare su prestazioni, disponibilità e costi.
- puoi prevedere i requisiti di capacità per controllare i costi.

Confronto tra caching serverless e cluster progettati internamente

Funzionalità	Cache serverless	Cluster progettati autonomamente
Configurazione della cache	Crea una cache con un solo nome in meno di un minuto	Fornisce un controllo granulare sulla progettazione del cluster di cache. L'utente può scegliere il tipo di nodo, il numero di nodi e il posizionamento tra le zone di disponibilità AWS
Supportato ElastiCache per la versione Redis	ElastiCache per Redis versione 7.1 e successive	ElastiCache per Redis versione 4.0 e successive
Modalità cluster	Utilizza Redis solo in <code>cluster mode enabled</code> . I client Redis devono supportare la connessione <code>cluster mode enabled</code> a ElastiCache Serverless.	Può essere configurato per funzionare in modalità cluster abilitata o in modalità cluster disabilitata.
Dimensionamento	Scalabilità automatica sia in verticale che in orizzontale senza alcuna gestione della capacità.	Fornisce il controllo sulla scalabilità e richiede al contempo il monitoraggio per assicurarsi che la capacità attuale soddisfi adeguatamente la domanda. È possibile scegliere di scalare verticalmente, aumentando o diminuendo la dimensione e del nodo di cache quando necessario. Puoi anche ridimensionare orizzontalmente, aggiungendo nuovi

Funzionalità	Cache serverless	Cluster progettati autonomamente
		<p>shard o aggiungendo altre repliche ai tuoi shard.</p> <p>Con la funzione Auto-Scaling puoi anche configurare il ridimensionamento in base a una pianificazione o in base a metriche come l'utilizzo della CPU e della memoria nella cache.</p>
Connessione client	I client si connettono a un singolo endpoint. Ciò consente di modificare la topologia del nodo di cache sottostante (scalabilità, sostituzioni e aggiornamenti) senza disconnettere il client.	I client si connettono a ogni singolo nodo di cache. Se un nodo viene sostituito, il client riscopre la topologia del cluster e ristabilisce le connessioni.
Configurabilità	Nessuna configurazione dettagliata disponibile. I clienti possono configurare le impostazioni di base, tra cui le sottoreti che possono accedere alla cache, l'attivazione o la disattivazione dei backup automatici e i limiti massimi di utilizzo della cache.	I cluster progettati autonomamente offrono opzioni di configurazione dettagliate. I clienti possono utilizzare gruppi di parametri per un controllo granulare. Per una tabella di questi valori di parametro per tipo di nodo, consulta Parametri specifici del tipo di nodo di Redis .

Funzionalità	Cache serverless	Cluster progettati autonomamente
Multi-AZ	I dati vengono replicati in modo asincrono su più zone di disponibilità per una maggiore disponibilità e una migliore latenza di lettura.	Offre la possibilità di progettare e il cluster in una singola zona di disponibilità o su più zone di disponibilità (AZ). Per i cluster Multi-AZ, i dati vengono replicati in modo asincrono su più zone di disponibilità per una maggiore disponibilità e una migliore latenza di lettura.
Crittografia a riposo	Sempre abilitato. I clienti possono utilizzare una chiave Chiave gestita da AWS o una chiave gestita dal cliente AWS KMS.	Opzione per abilitare o disabilitare la crittografia a riposo. Se abilitata, i clienti possono utilizzare una chiave Chiave gestita da AWS o una chiave gestita dal cliente AWS KMS.
Crittografia in transito (TLS)	Sempre abilitata. I clienti devono supportare la connettività TLS.	Opzione per abilitare o disabilitare.

Funzionalità	Cache serverless	Cluster progettati autonomamente
Backup	<p>Supporta backup automatici e manuali delle cache senza impatto sulle prestazioni.</p> <p>I backup sono interoperabili e possono essere ripristinati in una cache ElastiCache Serverless o in un cluster progettato autonomamente.</p>	<p>Supporta backup automatici e manuali. I cluster possono subire un impatto sulle prestazioni a seconda della memoria riservata disponibile. Per ulteriori informazioni, consulta Gestione della memoria prenotata.</p> <p>I backup sono interoperabili e possono essere ripristinati in una cache ElastiCache Serverless o in un cluster progettato autonomamente.</p>
Monitoraggio	<p>Supporta le metriche a livello di cache, tra cui la frequenza di accesso alla cache, la percentuale di errori della cache, la dimensione dei dati e le ECPU utilizzate.</p> <p>ElastiCache Serverless invia eventi utilizzando EventBridge quando si verificano eventi significativi nella cache. Puoi scegliere di monitorare, importare, trasformare e intervenire sugli ElastiCache eventi utilizzando Amazon EventBridge. Per ulteriori informazioni, consulta Eventi di cache serverless.</p>	<p>ElastiCache i cluster progettati autonomamente emettono parametri a ogni livello di nodo, inclusi parametri a livello di host e parametri di cache.</p> <p>I cluster progettati autonomamente emettono notifiche SNS per eventi significativi. Per informazioni, consulta Parametri di Redis.</p>

Funzionalità	Cache serverless	Cluster progettati autonomamente
Disponibilità	Contratto sul livello di servizio (SLA) di disponibilità del 99,99%	I cluster progettati autonomamente possono essere progettati per raggiungere un accordo sul livello di servizio (SLA) di disponibilità fino al 99,99%, a seconda della configurazione.
Aggiornamenti e patch del software	Aggiorna automaticamente il software di cache all'ultima versione secondaria e alla patch più recente, senza impatto sull'applicazione. I clienti ricevono una notifica per gli aggiornamenti delle versioni principali e possono eseguire l'aggiornamento alla versione principale più recente quando lo desiderano.	I cluster progettati autonomamente offrono un servizio self-service abilitato dal cliente per gli aggiornamenti delle versioni minori e corredati di patch, nonché per gli aggiornamenti delle versioni principali. Gli aggiornamenti gestiti vengono applicati automaticamente durante le finestre di manutenzione definite dal cliente. I clienti possono anche scegliere di applicare un aggiornamento di una versione secondaria o di una patch su richiesta.
Archivio dati globale	Non supportato	Supporta Global Data Store, che consente la replica interregionale con scritture su una singola regione e letture multiregionali

Funzionalità	Cache serverless	Cluster progettati autonomamente
Suddivisione dei dati su più livelli	Non supportato	I cluster progettati utilizzando nodi della famiglia r6gd hanno i dati suddivisi su più livelli tra la memoria e lo storage SSD locale (unità a stato solido). Il data tiering offre un'opzione in termini di rapporto prezzo/prestazioni per i carichi di lavoro Redis grazie all'utilizzo di unità a stato solido (SSD) a basso costo in ogni nodo del cluster, oltre all'archiviazione dei dati in memoria.
Modello tariffario	Pay-per-use, in base ai dati archiviati in GB all'ora e alle richieste nelle unità di elaborazione (ECPU). ElastiCache Per i dettagli dei prezzi, consulta questa pagina .	Pay-per-hour, in base all'utilizzo del nodo cache. Per i dettagli dei prezzi, consulta questa pagina .

Argomenti correlati:

- [Progettazione e gestione del proprio cluster ElastiCache](#)

Risorse Amazon ElastiCache

Ti consigliamo di iniziare a leggere le seguenti sezioni e di farvi riferimento ogni volta che ti servono:

- Punti salienti del servizio e prezzi - Nella [pagina sui dettagli del prodotto](#) trovi la panoramica prodotto, i punti salienti del servizio e i prezzi di ElastiCache.

- Video di ElastiCache – La sezione [Video relativi a ElastiCache](#) contiene video che presentano Amazon ElastiCache. I video illustrano casi d'uso comuni di ElastiCache e dimostrano come utilizzare ElastiCache per ridurre la latenza e migliorare la velocità effettiva per le applicazioni.
- Nozione di base – La [Guida introduttiva ad Amazon ElastiCache for Redis](#) sezione include informazioni sulla creazione di un cluster di cache. L'esempio descrive inoltre come autorizzare l'accesso al cluster cache, come connettere un nodo cache e come eliminare il cluster cache.
- Prestazioni su scala— Il whitepaper [Prestazioni su scala con Amazon ElastiCache](#) affronta le strategie di memorizzazione nella cache che aiutano l'applicazione a funzionare bene su larga scala.

Se vuoi utilizzare l'AWS Command Line Interface (AWS CLI), puoi utilizzare questi documenti per iniziare:

- [Documentazione di AWS Command Line Interface](#)

In questa sezione vengono fornite informazioni sul download della AWS CLI, come fare in modo che la AWS CLI lavori sul tuo sistema e fornendo le tue credenziali AWS.

- [Documentazione relativa alla AWS CLI per ElastiCache](#)

Questo documento distinto riguarda tutti i comandi della AWS CLI per ElastiCache, inclusi la sintassi e gli esempi.

Puoi scrivere programmi applicativi per utilizzare l'API di ElastiCache con una serie di linguaggi di programmazione di uso comune. Ecco alcune risorse:

- [Strumenti per Amazon Web Services](#)

Amazon Web Services offre numerosi Software Development Kit (SDK) che supportano ElastiCache. Puoi scrivere codici per ElastiCache utilizzando Java, .NET, PHP, Ruby e altri linguaggi. Questi SDK possono semplificare enormemente lo sviluppo delle applicazioni formattando le richieste a ElastiCache, analizzando le risposte e offrendo logica di ripetizione e gestione degli errori.

- [Utilizzo dell'API ElastiCache](#)

Se non intendi utilizzare i SDK di AWS, puoi interagire direttamente con ElastiCache utilizzando l'API di query. In questa sezione puoi trovare suggerimenti per risolvere i problemi e informazioni su come creare e autenticare richieste e gestire risposte.

- [Informazioni sull'API di Amazon ElastiCache](#)

Questo documento distinto riguarda tutte le operazioni dell'API di ElastiCache, inclusi la sintassi e gli esempi.

AWSRegioni e zone di disponibilità

Le risorse di cloud computing Amazon sono ospitate in strutture dei data center disponibili in diverse aree nel mondo, ad esempio Nord America, Europa o Asia. La posizione di ogni data center è detta regione AWS.

Ogni regione AWS contiene diverse posizioni chiamate zone di disponibilità o AZ (Availability Zone). Ogni zona di disponibilità è progettata per rimanere isolata dai guasti che si verificano in altre zone di disponibilità. Ciascuna è progettata per fornire una connettività di rete non costosa e a bassa latenza ad altre zone di disponibilità nella stessa regione AWS. Avviando istanze in zone di disponibilità separate, potrai proteggere le tue applicazioni dai guasti di una singola posizione. Per ulteriori informazioni, consulta [Scelta di regioni e zone di disponibilità](#).

Grazie a un'opzione nota come implementazione Multi-AZ, è possibile creare il cluster in diverse zone di disponibilità. Quando scegli questa opzione, Amazon effettua automaticamente il provisioning e la gestione di un'istanza di nodo in standby secondaria situata in una zona di disponibilità diversa. L'istanza del nodo primario viene replicata in modo asincrono tra le zone di disponibilità nell'istanza secondaria. Questo approccio consente di fornire ridondanza dei dati e supporto per il failover, eliminare blocchi I/O e ridurre al minimo i picchi di latenza durante i backup di sistema. Per ulteriori informazioni, vedere [Riduzione al minimo dei tempi di inattività in ElastiCache per Redis: Multi-AZ](#).

Casi d'uso comuni relativi a ElastiCache e in che modo ElastiCache può essere d'aiuto

Che si tratti di fornire le ultime notizie del giorno, una classifica dei primi 10 punteggi di gioco, un catalogo di prodotti o che si tratti di vendere biglietti per un evento, la velocità è alla base di ogni operazione. Il successo del tuo sito web e del tuo business è influenzato significativamente dalla velocità con cui si distribuiscono i contenuti.

Nell'articolo "[For Impatient Web Users, an Eye Blink Is Just Too Long to Wait \(Per gli utenti del web impazienti, attendere un battito di ciglia è troppo\)](#)", il New York Times ha stimato che gli utenti possono registrare una differenza di 250 millisecondi (1/4 di secondo) tra i siti concorrenti. Gli utenti tendono a scartare il sito più lento a favore di quello più veloce. Test effettuati da Amazon, secondo quanto descritto in "[How Webpage Load Time Is Related to Visitor Loss \(Come il tempo di caricamento di una pagina Web è collegato alla perdita di visitatori\)](#)", hanno rivelato che ogni 100 ms in più (1/10 di secondo) impiegati nel caricamento della pagina, le vendite subiscono una flessione dell'1 per cento.

Se qualcuno desidera dei dati, è possibile fornirli in modo molto più veloce se sono memorizzati nella cache. Che si tratti di una pagina web o di un report alla base di decisioni aziendali. La tua azienda può permettersi di non memorizzare in cache le pagine web in modo da distribuirle con la minore latenza possibile?

Intuitivamente, potrebbe essere piuttosto scontato pensare di memorizzare in cache gli elementi richiesti con maggiore frequenza. Invece, perché non memorizzare in cache gli elementi richiesti con minore frequenza? Anche le chiamate API remote o le query di database meglio ottimizzate sono notevolmente più lente rispetto a quando si richiama una chiave flat da una cache in memoria. I tempi notevolmente più lenti spingono i clienti ad andare altrove.

I seguenti esempi mostrano alcuni dei modi in cui l'utilizzo di ElastiCache può migliorare le prestazioni generali dell'applicazione.

Argomenti

- [Datastore in memoria](#)
- [Classifiche per videogiochi \(set ordinati Redis\)](#)
- [Messaggistica \(pub/sub Redis\)](#)
- [Data consigliati \(hash redis\)](#)
- [Altri usi di Redis](#)

- [Testimonianze dei clienti su ElastiCache](#)

Datastore in memoria

Lo scopo principale di uno store chiave-valore in memoria è fornire accesso ultrarapido (latenza in millisecondi) e a costo zero alle copie di dati. La maggior parte dei datastore hanno aree di dati a cui si accede di frequente ma che vengono aggiornate raramente. Inoltre, l'interrogazione di un database è sempre più lenta e più costosa dell'individuazione di una chiave in una cache della coppia chiave-valore. Alcune query di database sono particolarmente costose da eseguire. Ad esempio, le query che implicano unioni di più tabelle o le query con calcoli complessi. Memorizzando nella cache tali risultati delle query, si paga il prezzo della query una sola volta. Quindi è possibile richiamare velocemente i dati più volte senza dover rieseguire la query.

Cosa devo memorizzare nella cache?

Quando si decide quali dati memorizzare nella cache, tenere presenti i seguenti fattori:

Velocità e spese – È sempre più lento e costoso ottenere dati da un database che da una cache. Alcune query del database sono per natura più lente e più costose di altre. Ad esempio, le query che eseguono unioni su più tabelle sono molto più lente e più costose delle semplici query a tabella unica. Se l'acquisizione di dati interessanti richiede una query più lenta e più costosa, è indicata per il caching. Se l'acquisizione di dati richiede una query semplice e relativamente rapida, può essere idonea per il caching ma è necessario valutare altri fattori.

Dati e modello di accesso— Determinare cosa memorizzare nella cache comporta anche la comprensione dei dati stessi e dei relativi modelli di accesso. Ad esempio, non ha senso memorizzare nella cache i dati sottoposti a continue modifiche o a cui si accede raramente. Affinché la memorizzazione nella cache fornisca vantaggi reali, i dati devono essere relativamente statici e l'accesso a essi deve essere frequente. Un esempio è un profilo personale su un sito di social media. D'altra parte, non è necessario memorizzare nella cache se questa operazione non fornisce vantaggi economici e non migliora la velocità. Ad esempio, non ha senso memorizzare in cache le pagine web che restituiscono risultati di ricerca, perché le query e i risultati sono di solito univoci.

Obsolescenza – Per definizione, i dati memorizzati nella cache sono dati obsoleti. Anche se in determinate circostanze non sono obsoleti, dovrebbero sempre essere considerati e trattati come obsoleti. Per dire se i dati siano candidati al caching, devi stabilire la tolleranza dell'applicazione per i dati obsoleti.

L'applicazione potrebbe essere in grado di tollerare i dati obsoleti in un dato contesto, ma non in un altro. Supponi, ad esempio, che il sito fornisca un prezzo delle azioni quotato in borsa. I clienti potrebbero accettare una certa obsolescenza con una dichiarazione di non responsabilità secondo la quale i prezzi potrebbero essere n minuti in ritardo. Tuttavia, quando si offre il prezzo per lo stesso stock a un broker che effettua una vendita o un acquisto, sono necessari i dati in tempo reale.

Prendi in considerazione di memorizzare nella cache i dati in presenza di una delle seguenti condizioni:

- L'acquisizione dei dati è lenta o costosa se confrontata con il recupero dalla cache.
- Gli utenti accedono spesso ai dati.
- I dati rimangono relativamente gli stessi, o se cambiano rapidamente l'obsolescenza non è un grosso problema.

Per ulteriori informazioni, consulta gli argomenti seguenti:

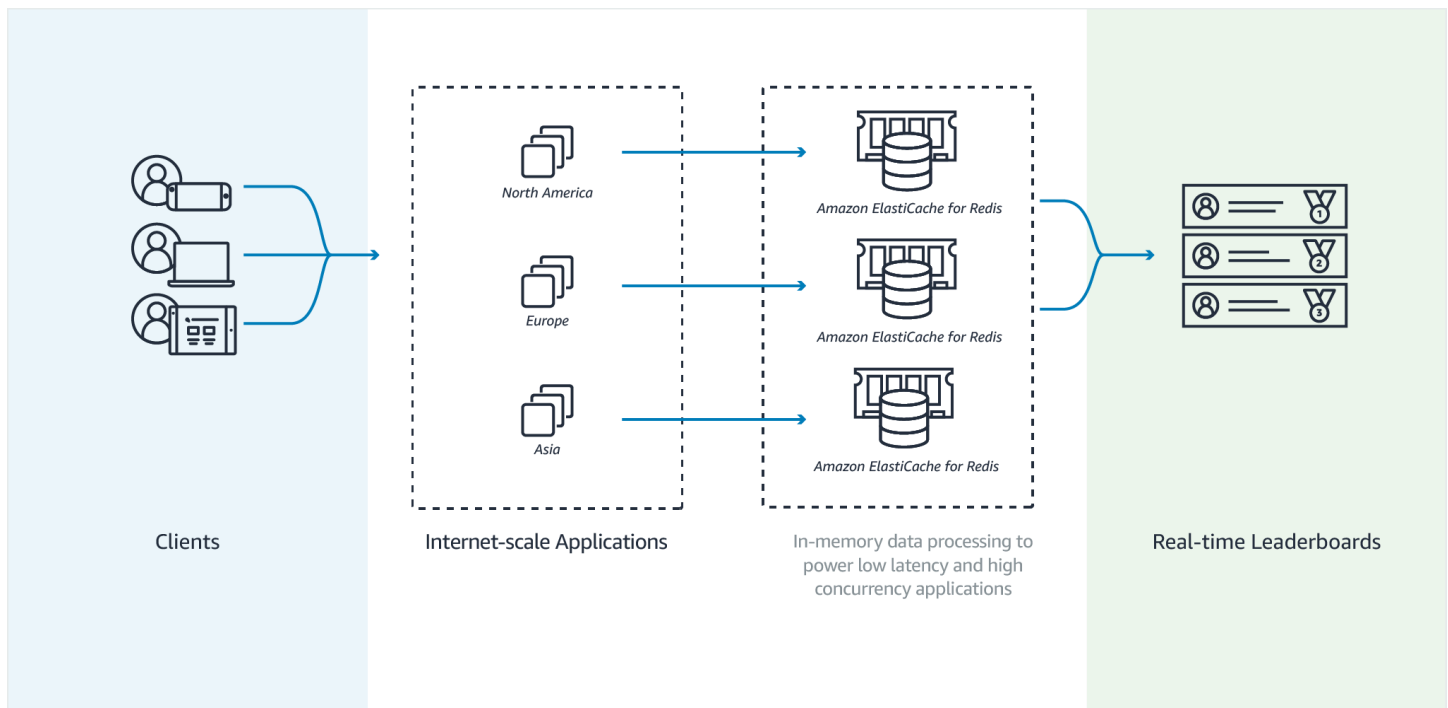
- [Strategie di caching](#) nella Guida per l'utente di ElastiCache for Redis

Classifiche per videogiochi (set ordinati Redis)

I set ordinati Redis spostano la complessità computazionale delle classifiche dall'applicazione al cluster Redis.

Le classifiche, ad esempio i primi 10 punteggi di una partita, sono complessi da un punto di vista computazionale. Ciò è particolarmente vero quando c'è un gran numero di giocatori concorrenti e punteggi che cambiano continuamente. I set ordinati Redis garantiscono unicità e ordine degli elementi. Utilizzando i set ordinati Redis, ogni volta che un nuovo elemento viene aggiunto al set ordinato, viene ricollocato nella classifica in tempo reale. Viene quindi aggiunto al set nel corretto ordine numerico.

Nel diagramma seguente, puoi vedere come funziona la classifica di gioco con ElastiCache for Redis.



Example – Classifiche Redis

In questo esempio, quattro giocatori e i relativi punteggi vengono inseriti in un elenco ordinato tramite ZADD. Il comando ZREVRANGEBYSCORE elenca i giocatori in base al punteggio, dal più alto al più basso. Quindi, ZADD viene utilizzato per aggiornare il punteggio di June sostituendo la voce esistente. Infine ZREVRANGEBYSCORE elenca i giocatori in base al punteggio, dal più alto al più basso. L'elenco mostra che June è salita di posizione nella classifica.

```
ZADD leaderboard 132 Robert
ZADD leaderboard 231 Sandra
ZADD leaderboard 32 June
ZADD leaderboard 381 Adam

ZREVRANGEBYSCORE leaderboard +inf -inf
1) Adam
2) Sandra
3) Robert
4) June

ZADD leaderboard 232 June

ZREVRANGEBYSCORE leaderboard +inf -inf
1) Adam
2) June
```

- 3) Sandra
- 4) Robert

Il comando seguente comunica a June qual è il suo posto nella classifica di tutti i giocatori. Poiché la classifica è basata su zero, ZREVRANK restituisce un 1 per June, che si trova in seconda posizione.

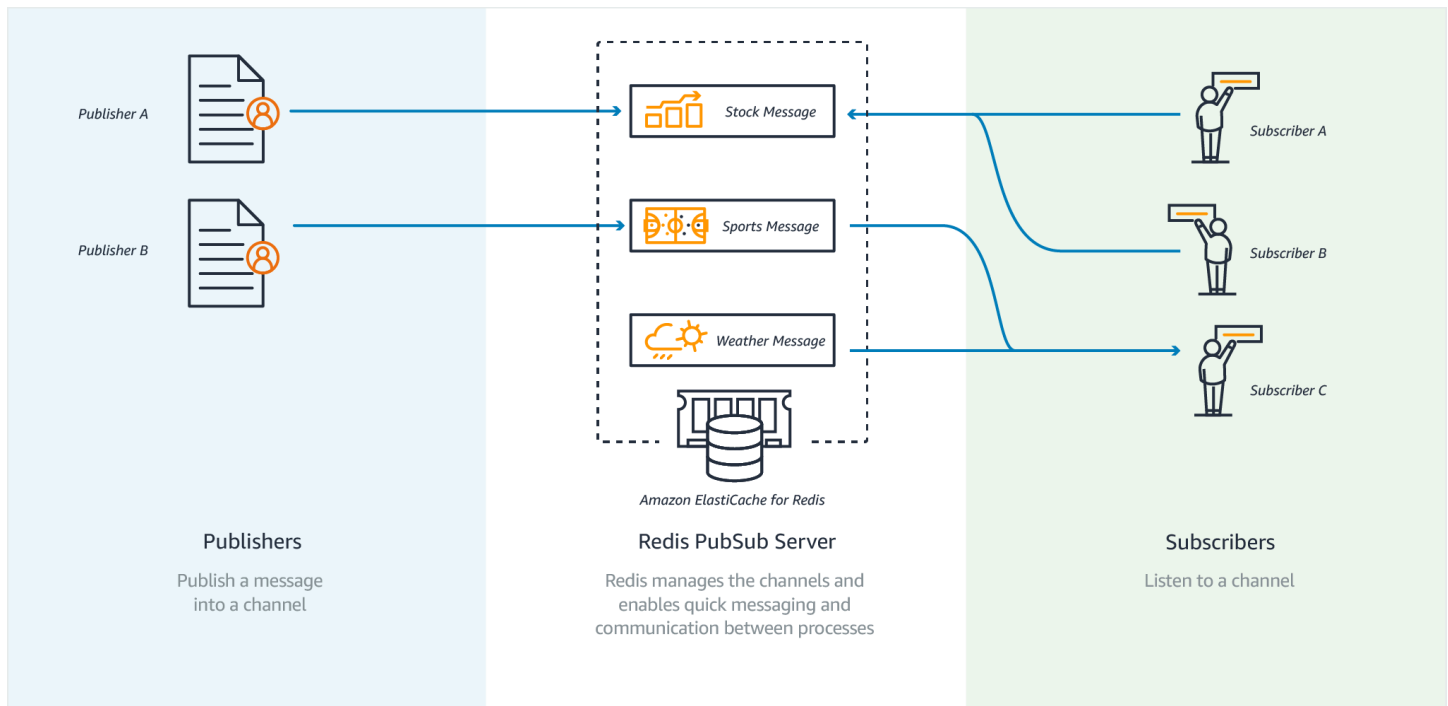
```
ZREVRANK leaderboard June
1
```

Per ulteriori informazioni, consulta la [documentazione Redis](#) sui set ordinati.

Messaggistica (pub/sub Redis)

Quando invii un messaggio e-mail, lo invii a uno o più destinatari specificati. Nel paradigma pub/sub, invii un messaggio a un canale specifico non sapendo chi eventualmente lo riceve. Le persone che ricevono il messaggio sono quelle che hanno effettuato la sottoscrizione al canale. Ad esempio, supponiamo che tu voglia effettuare la sottoscrizione al canale news.sports.golf: Tu e tutti gli iscritti al canale news.sports.golf ricevono tutti i messaggi pubblicati su news.sports.golf.

La caratteristica pub/sub Redis non ha relazione con nessuno spazio chiave. Perciò, non interferisce a nessun livello. Nel diagramma seguente, puoi trovare un'illustrazione della messaggistica di ElastiCache for Redis.



Sottoscrizione in corso

Per ricevere messaggi su un canale, effettua la sottoscrizione al canale. Puoi effettuare la sottoscrizione al canale, a più canali specificati o a tutti i canali che corrispondono a un modello. Per annullare una sottoscrizione, annulli la sottoscrizione dal canale specificato al momento dell'iscrizione. In alternativa, se la sottoscrizione è stata effettuata utilizzando la corrispondenza dei modelli, la si annulla utilizzando lo stesso modello utilizzato in precedenza.

Example – Effettuare una sottoscrizione a un singolo canale

Per effettuare una sottoscrizione a un singolo canale, utilizza il comando UNSUBSCRIBE specificando il canale a cui hai effettuato la sottoscrizione. Nell'esempio seguente, un cliente effettua la sottoscrizione al canale news.sports.golf.

```
SUBSCRIBE news.sports.golf
```

Trascorso un certo periodo, il cliente annulla la sottoscrizione al canale utilizzando il comando UNSUBSCRIBE specificando il canale dal quale annullare la sottoscrizione.

```
UNSUBSCRIBE news.sports.golf
```

Example – Sottoscrizioni a più canali specificati

Per annullare una sottoscrizione a più canali specificati, elenca i canali con il comando SUBSCRIBE. Nell'esempio seguente, un cliente effettua la sottoscrizione ai canali news.sports.golf, news.sports.soccer e news.sports.skiing.

```
SUBSCRIBE news.sports.golf news.sports.soccer news.sports.skiing
```

Per annullare la sottoscrizione a un canale specificato, utilizza il comando UNSUBSCRIBE specificando il canale dal quale annullare la sottoscrizione.

```
UNSUBSCRIBE news.sports.golf
```

Per annullare una sottoscrizione a più canali, utilizza il comando UNSUBSCRIBE e specifica i canali dai quali annullare la sottoscrizione.

```
UNSUBSCRIBE news.sports.golf news.sports.soccer
```


Per annullare tutte le sottoscrizioni, usa UNSUBSCRIBE e specifica ogni canale. Oppure usa UNSUBSCRIBE e non specificare un canale.

```
UNSUBSCRIBE news.sports.golf news.sports.soccer news.sports.skiing
```

oppure

```
UNSUBSCRIBE
```

Example – Sottoscrizioni mediante la corrispondenza di modelli

I clienti possono effettuare la sottoscrizione a tutti i canali corrispondenti a un modello mediante il comando PSUBSCRIBE.

Nell'esempio seguente, un cliente effettua la sottoscrizione a tutti i canali sportivi. Non elenchi tutti i canali sportivi singolarmente, come si farebbe con SUBSCRIBE. Invece, con il comando PSUBSCRIBE si utilizza la corrispondenza del modello.

```
PSUBSCRIBE news.sports.*
```

Example Annullamento delle sottoscrizioni

Per annullare le sottoscrizioni a questi canali, utilizza il comando PUNSUBSCRIBE.

```
PUNSUBSCRIBE news.sports.*
```

Important

Le stringhe del canale inviate ai comandi [P]SUBSCRIBE e [P]UNSUBSCRIBE devono corrispondere. Non è possibile PSUBSCRIBE a news.* e PUNSUBSCRIBE da news.sports.* o UNSUBSCRIBE da news.sports.golf.

Pubblicazione

Per inviare un messaggio a tutti coloro che hanno effettuato la sottoscrizione a un canale, utilizza il comando PUBLISH specificando il canale e il messaggio. L'esempio seguente pubblica il messaggio "It's Saturday and sunny (è sabato e fa bel tempo. I'm headed to the links. (Vado ai link)". sul canale news.sports.golf.

```
PUBLISH news.sports.golf "It's Saturday and sunny. I'm headed to the links."
```

Un cliente non può pubblicare sul canale al quale ha effettuato la sottoscrizione.

Per ulteriori informazioni, consulta [Pub/Sub](#) nella documentazione Redis.

Data consigliati (hash redis)

L'utilizzo di INCR e DECR semplifica i suggerimenti di compilazione. Ogni volta che un utente aggiunge un "mi piace" a un prodotto, si incrementa il contatore `item:productID:like`. Ogni volta che un utente aggiunge un "non mi piace" a un prodotto, si incrementa il contatore `item:productID:dislike`. L'utilizzo di hash Redis ti consente inoltre di mantenere un elenco di tutti coloro i quali hanno aggiunto un "mi piace" e un "non mi piace" a un prodotto.

Example - "Mi piace" e "Non mi piace"

```
INCR item:38923:likes  
HSET item:38923:ratings Susan 1  
INCR item:38923:dislikes  
HSET item:38923:ratings Tommy -1
```

Altri usi di Redis

Il blog post [How to take advantage of Redis just adding it to your stack \(Come sfruttare Redis semplicemente aggiungendolo al tuo mucchio\)](#) di Salvatore Sanfilippo parla di una serie di preoccupazioni comuni riguardanti i database e come possono essere facilmente risolte con Redis. Questo approccio snellisce il database e migliora le prestazioni.

Testimonianze dei clienti su ElastiCache

Per scoprire come aziende del calibro di Airbnb, PBS, Esri e molte altre utilizzano Amazon ElastiCache per far crescere il loro business con un'esperienza cliente potenziata, consulta [Modi di utilizzo di ElastiCache da parte dei clienti](#).

Puoi anche guardare i [video tutorial](#) per altri casi d'uso dei clienti ElastiCache.

Guida introduttiva ad Amazon ElastiCache for Redis

Usa il tutorial pratico in questa sezione per aiutarti a iniziare e saperne di più su Redis ElastiCache .

Argomenti

- [Configurazione](#)
- [Fase 1: creazione di una cache](#)
- [Fase 2: lettura e scrittura di dati nella cache](#)
- [Fase 3: \(facoltativo\) pulizia](#)
- [Fasi successive](#)
- [Endamenti su ElastiCache e AWS SDK](#)
- [Tutorial: Configurazione di una funzione Lambda per accedere ad Amazon in un ElastiCache Amazon VPC](#)

Configurazione

Per configurare ElastiCache:

Argomenti

- [Registrati per un Account AWS](#)
- [Creazione di un utente amministratore](#)
- [Concessione dell'accesso programmatico](#)
- [Configura le tue autorizzazioni \(solo nuovi ElastiCache utenti\)](#)
- [Configurazione di EC2](#)
- [Assegnazione dell'accesso di rete di un gruppo di sicurezza Amazon VPC alla cache](#)
- [Download e configurazione di redis-cli](#)

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 best practice di sicurezza, [assegna l'accesso amministrativo a un utente amministrativo](#) e utilizza solo l'utente root per eseguire [attività che richiedono l'accesso di un utente root](#).

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. 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.

Creazione di un utente amministratore

Dopo la registrazione 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 [Accesso come utente root](#) 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.

Creazione di un utente amministratore

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 Centro identità AWS IAM, assegna l'accesso amministrativo a un utente amministrativo.

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

Accesso come utente amministratore

- Per accedere 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.

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

Quale utente necessita dell'accesso programmatico?	Per	Come
		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 riferimento agli SDK e agli AWS strumenti. • 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 ElastiCache utenti)

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.

Amazon ElastiCache crea e utilizza ruoli collegati ai servizi per fornire risorse e accedere ad altre AWS risorse e servizi per tuo conto. Per creare ElastiCache per te un ruolo collegato ai servizi, utilizza la policy -managed denominata AWS. AmazonElastiCacheFullAccess 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, verificare di avere le autorizzazioni per chiamare `iam:createServiceLinkedRole` o per creare il ruolo collegato ai servizi ElastiCache.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Creare una nuova policy](#) (IAM)
- [Policy gestite da AWS per Amazon ElastiCache](#)
- [Utilizzo di ruoli collegati ai servizi per Amazon ElastiCache](#)

Configurazione di EC2

Dovrai configurare un'istanza EC2 da cui potrai connetterti alla cache.

- Se non disponi già di un'istanza EC2, le informazioni per configurare un'istanza EC2 sono disponibili in [Tutorial: Get started with Amazon EC2 Linux instances](#).
- L'istanza EC2 deve trovarsi nello stesso VPC e avere le stesse impostazioni del gruppo di sicurezza della cache. Per impostazione predefinita, Amazon ElastiCache crea una cache nel tuo VPC predefinito e utilizza il gruppo di sicurezza predefinito. Per questo tutorial, verifica che la tua istanza EC2 si trovi nel VPC predefinito e disponga del gruppo di sicurezza predefinito.

Assegnazione dell'accesso di rete di un gruppo di sicurezza Amazon VPC alla cache

ElastiCache i cluster progettati autonomamente utilizzano la porta 6379 per i comandi Redis e la modalità ElastiCache serverless utilizza sia la porta 6379 che la porta 6380. Per connettere ed eseguire correttamente i comandi Redis dall'istanza EC2, il gruppo di sicurezza deve consentire l'accesso a queste porte secondo necessità.

1. Accedi AWS Command Line Interface e apri la console [Amazon EC2](#).
2. Nel riquadro di navigazione, in Rete e sicurezza, scegli Gruppi di sicurezza.
3. Dall'elenco dei gruppi di sicurezza, scegli il gruppo di sicurezza per il VPC Amazon. A meno che tu non abbia creato un gruppo di sicurezza da ElastiCache utilizzare, questo gruppo di sicurezza verrà denominato predefinito.
4. Seleziona la scheda In entrata, quindi:
 - a. Scegli Modifica.
 - b. Scegli Aggiungi regola.
 - c. Nella colonna Tipo, scegli Regola TCP personalizzata.
 - d. Nella casella Intervallo porte digita 6379.
 - e. Nella casella Origine scegli Ovunque che ha l'intervallo di porte (0.0.0.0/0) in modo che qualunque istanza Amazon EC2 avviata in Amazon VPC possa connettersi alla cache.
 - f. Se utilizzi ElastiCache serverless, aggiungi un'altra regola scegliendo Aggiungi regola.
 - g. Nella colonna Tipo, scegli Regola TCP personalizzata.
 - h. Nella casella Intervallo porte digita 6380.

- i. Nella casella Origine scegli Ovunque che ha l'intervallo di porte (0.0.0.0/0) in modo che qualunque istanza Amazon EC2 avviata in Amazon VPC possa connettersi alla cache.
- j. Seleziona Salva

Download e configurazione di redis-cli

1. Connettersi all'istanza Amazon EC2 utilizzando l'utilità di connessione preferita. Per istruzioni su come connetterti a un'istanza Amazon EC2, consulta la [Guida alle operazioni di base di Amazon EC2](#).
2. Scarica e installa l'utilità redis-cli eseguendo il comando appropriato per la configurazione.

Amazon Linux 2023

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

Amazon Linux 2

```
sudo amazon-linux-extras install epel -y
sudo yum install gcc jemalloc-devel openssl-devel tcl tcl-devel -y
sudo wget http://download.redis.io/redis-stable.tar.gz
sudo tar xvzf redis-stable.tar.gz
cd redis-stable
sudo make BUILD_TLS=yes
```

Note

- Quando installi il pacchetto redis6, viene installato redis6-cli con il supporto di crittografia predefinito.
- È importante disporre del supporto integrato per TLS durante l'installazione di redis-cli. ElastiCache Serverless è accessibile solo quando TLS è abilitato.
- Se ci si connette a un cluster non crittografato, non è necessaria l'opzione `Build_TLS=yes`.

Fase 1: creazione di una cache

In questa fase crei una nuova cache in Amazon ElastiCache.

AWS Management Console

Per creare una nuova cache tramite la console ElastiCache:

1. Accedi alla AWS Management Console e apri la console all'indirizzo <https://console.aws.amazon.com/connect/>.
2. Nel riquadro di navigazione sul lato sinistro della console scegli Cache Redis.
3. Sul lato destro della console seleziona Crea cache Redis.
4. In Impostazioni cache inserisci un nome. Facoltativamente, è possibile inserire una descrizione per la cache.
5. Lascia le impostazioni predefinite selezionate.
6. Fai clic su Crea per creare la cache.
7. Una volta che lo stato della cache è "ATTIVO", puoi iniziare a scrivere e leggere i dati nella cache.

AWS CLI

L'esempio della AWS CLI seguente crea una nuova cache utilizzando create-serverless-cache.

Linux

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine redis
```

Windows

```
aws elasticache create-serverless-cache ^  
  --serverless-cache-name CacheName ^  
  --engine redis
```

Il valore del campo Stato è impostato su CREATING.

Per verificare se ElastiCache ha terminato la creazione della cache, utilizza il comando `describe-serverless-caches`.

Linux

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches --serverless-cache-name CacheName
```

Dopo aver creato la nuova cache, procedi alla [Fase 2: lettura e scrittura di dati nella cache](#).

Fase 2: lettura e scrittura di dati nella cache

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](#).

Questa sezione presuppone inoltre che tu abbia configurato l'accesso al VPC e le impostazioni del gruppo di sicurezza per l'istanza EC2 da cui ti connetti alla cache e configurato `redis-cli` sull'istanza EC2. Per ulteriori informazioni su questa fase, consulta [Configurazione](#).

Ricerca dell'endpoint della cache

AWS Management Console

Per trovare l'endpoint della cache tramite la console ElastiCache:

1. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione sul lato sinistro della console scegli Cache Redis.
3. Sul lato destro della console fai clic sul nome della cache che hai appena creato.
4. In Dettagli della cache individua e copia l'endpoint della cache.

AWS CLI

L'esempio della AWS CLI seguente mostra come trovare l'endpoint per la nuova cache utilizzando il comando `describe-serverless-caches`. Dopo aver eseguito il comando, cerca il campo "Endpoint".

Linux

```
aws elasticache describe-serverless-caches \  
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache describe-serverless-caches ^  
  --serverless-cache-name CacheName
```

Connessione alla cache Redis (Linux)

Ora che disponi dell'endpoint necessario, puoi accedere a un'istanza EC2 e connetterti alla cache. Nell'esempio seguente, viene utilizzata l'utilità `redis-cli` per eseguire la connessione a un cluster che esegue Redis. Il comando seguente esegue la connessione a una cache (nota: sostituisci `cache-endpoint` con l'endpoint recuperato nella fase precedente).

```
src/redis-cli -h cache-endpoint --tls -p 6379  
set a "hello"           // Set key "a" with a string value and no expiration  
OK  
get a                   // Get value for key "a"  
"hello"
```

Connessione alla cache Redis (Windows)

Ora che disponi dell'endpoint necessario, puoi accedere a un'istanza EC2 e connetterti alla cache. Nell'esempio seguente, viene utilizzata l'utilità `redis-cli` per eseguire la connessione a un cluster che esegue Redis. Il seguente comando esegue la connessione a una cache. Apri il prompt dei comandi, passa alla directory Redis ed esegui il comando (nota: sostituisci `Cache_Endpoint` con l'endpoint recuperato nella fase precedente).

```
c:\Redis>redis-cli -h Redis_Cluster_Endpoint --tls -p 6379  
set a "hello"           // Set key "a" with a string value and no expiration  
OK  
get a                   // Get value for key "a"  
"hello"
```

Ora puoi procedere alla [Fase 3: \(facoltativo\) pulizia](#).

Fase 3: (facoltativo) pulizia

Se non hai più bisogno della cache Amazon ElastiCache creata, puoi eliminarla. In questo modo hai la certezza che non ti vengano addebitati costi per risorse che non stai utilizzando. È possibile utilizzare la console ElastiCache, la AWS CLI o l'API ElastiCache per eliminare la cache.

AWS Management Console

Per eliminare la cache tramite la console:

1. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione sul lato sinistro della console scegli Cache Redis.
3. Scegli il pulsante di opzione accanto alla cache che vuoi eliminare.
4. Seleziona Operazioni in alto a destra e scegli Elimina.
5. Facoltativamente, puoi scegliere di acquisire uno snapshot finale prima di eliminare la cache.
6. Nella schermata di conferma Elimina reinserisci il nome della cache e scegli Elimina per eliminare la cache o Annulla per mantenerla.

Non appena la cache passa allo stato ELIMINAZIONE IN CORSO, non vengono più addebitati i costi.

AWS CLI

L'esempio della AWS CLI seguente elimina una cache utilizzando il comando `delete-serverless-cache`.

Linux

```
aws elasticache delete-serverless-cache \  
  --serverless-cache-name CacheName
```

Windows

```
aws elasticache delete-serverless-cache ^  
  --serverless-cache-name CacheName
```

Il valore del campo Stato è impostato su ELIMINAZIONE IN CORSO.

Ora puoi procedere alla [Fasi successive](#).

Fasi successive

Per ulteriori informazioni su ElastiCache, consulta le pagine seguenti:

- [Lavorare con ElastiCache](#)
- [Scalabilità per Redis ElastiCache](#)
- [Registrazione di log e monitoraggio in Amazon ElastiCache](#)
- [Best practice e strategie di caching ElastiCache](#)
- [Snapshot e ripristino](#)
- [Monitoraggio di Amazon SNS per gli eventi ElastiCache](#)

Endamenti su ElastiCache e AWS SDK

Questa sezione contiene tutorial pratici per aiutarti a saperne di più su Amazon ElastiCache. Ti raccomandiamo di lavorare con dei tutorial specifici della lingua.

Note

I servizi SDK AWS sono disponibili per varie lingue. Per un elenco completo consulta la pagina relativa agli [Strumenti per Amazon Web Services](#).

Python e ElastiCache

In questo tutorial utilizzerai il servizio SDK for Python (Boto3) AWS per scrivere semplici programmi che eseguono le seguenti operazioni ElastiCache:

- Creare cluster ElastiCache (modalità cluster abilitata e modalità cluster disabilitata)
- Controlla se esistono utenti o gruppi di utenti, altrimenti creali (solo Redis 6.0 e versioni successive)
- Connect in ElastiCache
- Eseguire operazioni come l'impostazione e l'acquisizione di stringhe, la lettura e la scrittura su streams e la pubblicazione e l'iscrizione dal canale Pub/Sub.

Man mano che procedi in questo tutorial, puoi fare riferimento alla documentazione AWS SDK for Python (Boto). Questa sezione è specifica di ElastiCache: [ElastiCache low-level client](#)

Prerequisiti dei tutorial

- Configurare una chiave di accesso AWS per utilizzare gli SDK AWS. Per ulteriori informazioni, consulta [Configurazione](#).
- Installare Python 3.0 o versione successiva. Per ulteriori informazioni, consulta <https://www.python.org/downloads>. Per istruzioni, consulta [Quickstart](#) nella documentazione di Boto 3.

Creazione di ElastiCache cluster e utenti

Gli esempi seguenti utilizzano l'SDK boto3 per le operazioni di ElastiCache gestione (creazione di cluster o utenti) e redis-py-cluster redis-py/ per la gestione dei dati.

Argomenti

- [Creare un cluster in modalità cluster disabilitata](#)
- [Creare un cluster in modalità cluster disabilitata con TLS e RBAC](#)
- [Creare un cluster in modalità cluster abilitata](#)
- [Creare un cluster in modalità cluster abilitata con TLS e RBAC](#)
- [Controllare che esistano utenti/gruppi di utenti, altrimenti crearli](#)

Creare un cluster in modalità cluster disabilitata

Copia il seguente programma e incollalo in un file denominato .py.

CreateClusterModeDisabledCluster

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
    create_cluster_mode_disabled(CacheNodeType='cache.t3.small', EngineVersion='6.0', NumCacheClusters=1,
    cache_cluster', ReplicationGroupId=None):
    """Creates an ElastiCache Cluster with cluster mode disabled
```

Returns a dictionary with the API response

:param CacheNodeType: Node type used on the cluster. If not specified, cache.t3.small will be used

Refer to <https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/CacheNodes.SupportedTypes.html> for supported node types

:param EngineVersion: Engine version to be used. If not specified, latest will be used.

:param NumCacheClusters: Number of nodes in the cluster. Minimum 1 (just a primary node) and maximum 6 (1 primary and 5 replicas).

If not specified, cluster will be created with 1 primary and 1 replica.

:param ReplicationGroupDescription: Description for the cluster.

:param ReplicationGroupId: Name for the cluster

:return: dictionary with the API results

"""

if not ReplicationGroupId:

 return 'ReplicationGroupId parameter is required'

response = client.create_replication_group(

 AutomaticFailoverEnabled=True,

 CacheNodeType=CacheNodeType,

 Engine='redis',

 EngineVersion=EngineVersion,

 NumCacheClusters=NumCacheClusters,

 ReplicationGroupDescription=ReplicationGroupDescription,

 ReplicationGroupId=ReplicationGroupId,

 SnapshotRetentionLimit=30,

)

return response

if __name__ == '__main__':

 # Creates an ElastiCache Cluster mode disabled cluster, based on cache.m6g.large nodes, Redis 6, one primary and two replicas

 elasticacheResponse = create_cluster_mode_disabled(

 #CacheNodeType='cache.m6g.large',

 EngineVersion='6.0',

 NumCacheClusters=3,

 ReplicationGroupDescription='Redis cluster mode disabled with replicas',

 ReplicationGroupId='redis202104053'

)


```
logging.info(elasticacheResponse)
```

Per eseguire il programma, immetti il comando seguente:

```
python CreateClusterModeDisabledCluster.py
```

Per ulteriori informazioni, consulta [Gestione dei cluster](#).

Creare un cluster in modalità cluster disabilitata con TLS e RBAC

Per garantire la sicurezza, è possibile utilizzare Transport Layer Security (TLS) e Controllo di accesso basato sul ruolo (RBAC) quando si crea un cluster disabilitato in modalità cluster. A differenza di Redis AUTH, in cui tutti i client autenticati dispongono di accesso completo al gruppo di replica se il token è autenticato, RBAC consente di controllare l'accesso al cluster tramite gruppi di utenti. Questi gruppi di utenti sono progettati come un modo per organizzare l'accesso ai gruppi di replica. Per ulteriori informazioni, consulta [Controllo accessi basato sui ruoli \(RBAC\)](#).

Copiare il seguente programma e incollarlo in un file denominato ClusterModeDisabledWithRBAC.py.

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
    create_cluster_mode_disabled_rbac(CacheNodeType='cache.t3.small', EngineVersion='6.0', NumCacheC
    cache cluster', ReplicationGroupId=None, UserGroupIds=None,
    SecurityGroupIds=None, CacheSubnetGroupName=None):
    """Creates an ElastiCache Cluster with cluster mode disabled and RBAC

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
    cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/
    CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
    used.
    :param NumCacheClusters: Number of nodes in the cluster. Minimum 1 (just a primary
    node) and maximum 6 (1 primary and 5 replicas).
    If not specified, cluster will be created with 1 primary and 1 replica.
    :param ReplicationGroupDescription: Description for the cluster.
```

```

:param ReplicationGroupId: Mandatory name for the cluster.
:param UserGroupIds: The ID of the user group to be assigned to the cluster.
:param SecurityGroupIds: List of security groups to be assigned. If not defined,
default will be used
:param CacheSubnetGroupName: subnet group where the cluster will be placed. If not
defined, default will be used.
:return: dictionary with the API results

"""
if not ReplicationGroupId:
    return {'Error': 'ReplicationGroupId parameter is required'}
elif not isinstance(UserGroupIds,(list)):
    return {'Error': 'UserGroupIds parameter is required and must be a list'}

params={'AutomaticFailoverEnabled': True,
        'CacheNodeType': CacheNodeType,
        'Engine': 'redis',
        'EngineVersion': EngineVersion,
        'NumCacheClusters': NumCacheClusters,
        'ReplicationGroupDescription': ReplicationGroupDescription,
        'ReplicationGroupId': ReplicationGroupId,
        'SnapshotRetentionLimit': 30,
        'TransitEncryptionEnabled': True,
        'UserGroupIds':UserGroupIds
        }

# defaults will be used if CacheSubnetGroupName or SecurityGroups are not explicit.
if isinstance(SecurityGroupIds,(list)):
    params.update({'SecurityGroupIds':SecurityGroupIds})
if CacheSubnetGroupName:
    params.update({'CacheSubnetGroupName':CacheSubnetGroupName})

response = client.create_replication_group(**params)
return response

if __name__ == '__main__':

    # Creates an ElastiCache Cluster mode disabled cluster, based on cache.m6g.large
nodes, Redis 6, one primary and two replicas.
    # Assigns the existent user group "mygroup" for RBAC authentication

    response=create_cluster_mode_disabled_rbac(
        CacheNodeType='cache.m6g.large',
        EngineVersion='6.0',

```

```
        NumCacheClusters=3,
        ReplicationGroupDescription='Redis cluster mode disabled with replicas',
        ReplicationGroupId='redis202104',
        UserGroupIds=[
            'mygroup'
        ],
        SecurityGroupIds=[
            'sg-7cc73803'
        ],
        CacheSubnetGroupName='default'
    )

logging.info(response)
```

Per eseguire il programma, immetti il comando seguente:

```
python ClusterModeDisabledWithRBAC.py
```

Per ulteriori informazioni, consulta [Gestione dei cluster](#).

Creare un cluster in modalità cluster abilitata

Copiare il seguente programma e incollarlo in un file denominato ClusterModeEnabled.py.

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
    create_cluster_mode_enabled(CacheNodeType='cache.t3.small', EngineVersion='6.0', NumNodeGroups=1,
    ReplicationGroupDescription='Sample cache with cluster mode
    enabled', ReplicationGroupId=None):
    """Creates an ElastiCache Cluster with cluster mode enabled

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
    cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/
    CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
    used.
```

```
:param NumNodeGroups: Number of shards in the cluster. Minimum 1 and maximum 90.
If not specified, cluster will be created with 1 shard.
:param ReplicasPerNodeGroup: Number of replicas per shard. If not specified 1
replica per shard will be created.
:param ReplicationGroupDescription: Description for the cluster.
:param ReplicationGroupId: Name for the cluster
:return: dictionary with the API results

"""
if not ReplicationGroupId:
    return 'ReplicationGroupId parameter is required'

response = client.create_replication_group(
    AutomaticFailoverEnabled=True,
    CacheNodeType=CacheNodeType,
    Engine='redis',
    EngineVersion=EngineVersion,
    ReplicationGroupDescription=ReplicationGroupDescription,
    ReplicationGroupId=ReplicationGroupId,
    # Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
node (implicit) and 2 replicas (replicasPerNodeGroup)
    NumNodeGroups=NumNodeGroups,
    ReplicasPerNodeGroup=ReplicasPerNodeGroup,
    CacheParameterGroupName='default.redis6.0.cluster.on'
)

return response

# Creates a cluster mode enabled
response = create_cluster_mode_enabled(
    CacheNodeType='cache.m6g.large',
    EngineVersion='6.0',
    ReplicationGroupDescription='Redis cluster mode enabled with replicas',
    ReplicationGroupId='redis20210',
    # Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
(implicit) and 2 replicas (replicasPerNodeGroup)
    NumNodeGroups=2,
    ReplicasPerNodeGroup=1,
)

logging.info(response)
```

Per eseguire il programma, immetti il comando seguente:

```
python ClusterModeEnabled.py
```

Per ulteriori informazioni, consulta [Gestione dei cluster](#).

Creare un cluster in modalità cluster abilitata con TLS e RBAC

Per garantire la sicurezza, è possibile utilizzare Transport Layer Security (TLS) e Controllo di accesso basato sul ruolo (RBAC) durante la creazione di un cluster in modalità cluster abilitata. A differenza di Redis AUTH, in cui tutti i client autenticati dispongono di accesso completo al gruppo di replica se il token è autenticato, RBAC consente di controllare l'accesso al cluster tramite gruppi di utenti. Questi gruppi di utenti sono progettati come un modo per organizzare l'accesso ai gruppi di replica. Per ulteriori informazioni, consulta [Controllo accessi basato sui ruoli \(RBAC\)](#).

Copiare il seguente programma e incollarlo in un file denominato ClusterModeEnabledWithRBAC.py.

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
client = boto3.client('elasticache')

def
    create_cluster_mode_enabled(CacheNodeType='cache.t3.small', EngineVersion='6.0', NumNodeGroups=1,
    ReplicationGroupDescription='Sample cache with cluster
mode enabled', ReplicationGroupId=None, UserGroupIds=None,
SecurityGroupIds=None, CacheSubnetGroupName=None, CacheParameterGroupName='default.redis6.0.clus
    """Creates an ElastiCache Cluster with cluster mode enabled and RBAC

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/
CacheNodes.SupportedTypes.html for supported node types
    :param EngineVersion: Engine version to be used. If not specified, latest will be
used.
    :param NumNodeGroups: Number of shards in the cluster. Minimum 1 and maximum 90.
If not specified, cluster will be created with 1 shard.
    :param ReplicasPerNodeGroup: Number of replicas per shard. If not specified 1
replica per shard will be created.
    :param ReplicationGroupDescription: Description for the cluster.
```

```

:param ReplicationGroupId: Name for the cluster.
:param CacheParameterGroupName: Parameter group to be used. Must be compatible with
the engine version and cluster mode enabled.
:return: dictionary with the API results

"""
if not ReplicationGroupId:
    return 'ReplicationGroupId parameter is required'
elif not isinstance(UserGroupIds,(list)):
    return {'Error': 'UserGroupIds parameter is required and must be a list'}

params={'AutomaticFailoverEnabled': True,
        'CacheNodeType': CacheNodeType,
        'Engine': 'redis',
        'EngineVersion': EngineVersion,
        'ReplicationGroupDescription': ReplicationGroupDescription,
        'ReplicationGroupId': ReplicationGroupId,
        'SnapshotRetentionLimit': 30,
        'TransitEncryptionEnabled': True,
        'UserGroupIds':UserGroupIds,
        'NumNodeGroups': NumNodeGroups,
        'ReplicasPerNodeGroup': ReplicasPerNodeGroup,
        'CacheParameterGroupName': CacheParameterGroupName
    }

# defaults will be used if CacheSubnetGroupName or SecurityGroups are not explicit.
if isinstance(SecurityGroupIds,(list)):
    params.update({'SecurityGroupIds':SecurityGroupIds})
if CacheSubnetGroupName:
    params.update({'CacheSubnetGroupName':CacheSubnetGroupName})

response = client.create_replication_group(**params)
return response

if __name__ == '__main__':
    # Creates a cluster mode enabled cluster
    response = create_cluster_mode_enabled(
        CacheNodeType='cache.m6g.large',
        EngineVersion='6.0',
        ReplicationGroupDescription='Redis cluster mode enabled with replicas',
        ReplicationGroupId='redis2021',
        # Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
        (implicit) and 2 replicas (replicasPerNodeGroup)
        NumNodeGroups=2,

```

```
        ReplicasPerNodeGroup=1,  
        UserGroupIds=[  
            'mygroup'  
        ],  
        SecurityGroupIds=[  
            'sg-7cc73803'  
        ],  
        CacheSubnetGroupName='default'  
    )  
  
    logging.info(response)
```

Per eseguire il programma, immetti il comando seguente:

```
python ClusterModeEnabledWithRBAC.py
```

Per ulteriori informazioni, consulta [Gestione dei cluster](#).

Controllare che esistano utenti/gruppi di utenti, altrimenti crearli

Con RBAC, è possibile creare utenti e assegnare loro autorizzazioni specifiche utilizzando una stringa di accesso. Gli utenti vengono assegnati a gruppi di utenti allineati a un ruolo specifico (amministratori, risorse umane) che vengono quindi distribuiti a uno o più gruppi di replica ElastiCache per Redis. In questo modo, è possibile stabilire limiti di sicurezza tra i client utilizzando lo stesso gruppo o gruppi di replica Redis e impedire ai client di accedere ai dati reciproci. Per ulteriori informazioni, consulta [Controllo accessi basato sui ruoli \(RBAC\)](#).

Copiate il seguente programma e incollatelo in un file denominato `.py`. `UserAndUserGroups` Aggiorna il meccanismo per fornire le credenziali. Le credenziali in questo esempio vengono mostrate come sostituibili e assegnate a un elemento non dichiarato. Evita le credenziali a codifica fissa.

```
import boto3  
import logging  
  
logging.basicConfig(level=logging.INFO)  
client = boto3.client('elasticache')  
  
def check_user_exists(UserId):  
    """Checks if UserId exists  
  
    Returns True if UserId exists, otherwise False  
    :param UserId: ElastiCache User ID
```

```
:return: True|False
"""
try:
    response = client.describe_users(
        UserId=UserId,
    )
    if response['Users'][0]['UserId'].lower() == UserId.lower():
        return True
except Exception as e:
    if e.response['Error']['Code'] == 'UserNotFound':
        logging.info(e.response['Error'])
        return False
    else:
        raise

def check_group_exists(UserGroupId):
    """Checks if UserGroupID exists

    Returns True if Group ID exists, otherwise False
    :param UserGroupId: ElastiCache User ID
    :return: True|False
    """

    try:
        response = client.describe_user_groups(
            UserGroupId=UserGroupId
        )
        if response['UserGroups'][0]['UserGroupId'].lower() == UserGroupId.lower():
            return True
    except Exception as e:
        if e.response['Error']['Code'] == 'UserGroupNotFound':
            logging.info(e.response['Error'])
            return False
        else:
            raise

def create_user(UserId=None,UserName=None>Password=None,AccessString=None):
    """Creates a new user

    Returns the ARN for the newly created user or the error message
    :param UserId: ElastiCache user ID. User IDs must be unique
    :param UserName: ElastiCache user name. ElastiCache allows multiple users with the
    same name as long as the associated user ID is unique.
    :param Password: Password for user. Must have at least 16 chars.
```



```
    :param AccessString: Access string with the permissions for the user. For
    details refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/
Clusters.RBAC.html#Access-string
    :return: user ARN
    """
    try:
        response = client.create_user(
            UserId=UserId,
            UserName=UserName,
            Engine='Redis',
            Passwords=[Password],
            AccessString=AccessString,
            NoPasswordRequired=False
        )
        return response['ARN']
    except Exception as e:
        logging.info(e.response['Error'])
        return e.response['Error']

def create_group(UserGroupId=None, UserIds=None):
    """Creates a new group.
    A default user is required (mandatory) and should be specified in the UserIds list

    Return: Group ARN
    :param UserIds: List with user IDs to be associated with the new group. A default
    user is required
    :param UserGroupId: The ID (name) for the group
    :return: Group ARN
    """
    try:
        response = client.create_user_group(
            UserGroupId=UserGroupId,
            Engine='Redis',
            UserIds=UserIds
        )
        return response['ARN']
    except Exception as e:
        logging.info(e.response['Error'])

if __name__ == '__main__':

    groupName='mygroup2'
    userName = 'myuser2'
```

```
userId=groupName+'-'+userName

# Creates a new user if the user ID does not exist.
for tmpUserId,tmpUserName in [ (userId,userName), (groupName+'-
default','default')]:
    if not check_user_exists(tmpUserId):
        response=create_user(UserId=tmpUserId,
UserName=EXAMPLE,Password=EXAMPLE,AccessString='on ~* +@all')
        logging.info(response)
    # assigns the new user ID to the user group
if not check_group_exists(groupName):
    UserIds = [ userId , groupName+'-default']
    response=create_group(UserGroupId=groupName,UserIds=UserIds)
    logging.info(response)
```

Per eseguire il programma, immetti il comando seguente:

```
python UserAndUserGroups.py
```

Connessione a ElastiCache

Negli esempi seguenti viene utilizzato il client Redis per connettersi a ElastiCache.

Argomenti

- [Connessione a un cluster non crittografato in modalità cluster disabilitata](#)
- [Connessione a un cluster abilitato in modalità cluster](#)

Connessione a un cluster non crittografato in modalità cluster disabilitata

Copia il seguente programma e incollalo in un file denominato ConnectClusterModeDisabled.py. Aggiorna il meccanismo per fornire le credenziali. Le credenziali in questo esempio vengono mostrate come sostituibili e assegnate a un elemento non dichiarato. Evita le credenziali a codifica fissa.

```
from redis import Redis
import logging

logging.basicConfig(level=logging.INFO)
redis = Redis(host='primary.xxx.yyyyyy.zzz1.cache.amazonaws.com', port=6379,
decode_responses=True, ssl=True, username=example, password=EXAMPLE)

if redis.ping():
```

```
logging.info("Connected to Redis")
```

Per eseguire il programma, immetti il comando seguente:

```
python ConnectClusterModeDisabled.py
```

Connessione a un cluster abilitato in modalità cluster

Copia il seguente programma e incollalo in un file denominato ConnectClusterModeEnabled.py.

```
from rediscluster import RedisCluster
import logging

logging.basicConfig(level=logging.INFO)
redis = RedisCluster(startup_nodes=[{"host":
    "xxx.yyy.clustercfg.zzz1.cache.amazonaws.com", "port": "6379"}],
    decode_responses=True, skip_full_coverage_check=True)

if redis.ping():
    logging.info("Connected to Redis")
```

Per eseguire il programma, immetti il comando seguente:

```
python ConnectClusterModeEnabled.py
```

Esempi di utilizzo

Gli esempi seguenti utilizzano l'SDK boto3 per ElastiCache per lavorare con ElastiCache.

Argomenti

- [Imposta e ottieni stringhe](#)
- [Imposta e ottieni un hash con più elementi](#)
- [Pubblicare \(scrivere\) e iscriversi \(leggere\) da un canale Pub/Sub](#)
- [Scrivere e leggere da un flusso](#)

Imposta e ottieni stringhe

Copia il seguente programma e incollalo in un file denominato SetAndGetStrings.py.

```
import time
import logging
```

```
logging.basicConfig(level=logging.INFO,format='%(asctime)s: %(message)s')

keyName='mykey'
currTime=time.ctime(time.time())

# Set the key 'mykey' with the current date and time as value.
# The Key will expire and removed from cache in 60 seconds.
redis.set(keyName, currTime, ex=60)

# Sleep just for better illustration of TTL (expiration) value
time.sleep(5)

# Retrieve the key value and current TTL
keyValue=redis.get(keyName)
keyTTL=redis.ttl(keyName)

logging.info("Key {} was set at {} and has {} seconds until expired".format(keyName,
    keyValue, keyTTL))
```

Per eseguire il programma, immetti il comando seguente:

```
python SetAndGetStrings.py
```

Imposta e ottieni un hash con più elementi

Copia il seguente programma e incollalo in un file denominato SetAndGetHash.py.

```
import logging
import time

logging.basicConfig(level=logging.INFO,format='%(asctime)s: %(message)s')

keyName='mykey'
keyValues={'datetime': time.ctime(time.time()), 'epochtime': time.time()}

# Set the hash 'mykey' with the current date and time in human readable format
# (datetime field) and epoch number (epochtime field).
redis.hset(keyName, mapping=keyValues)

# Set the key to expire and removed from cache in 60 seconds.
redis.expire(keyName, 60)

# Sleep just for better illustration of TTL (expiration) value
time.sleep(5)
```

```
# Retrieves all the fields and current TTL
keyValues=redis.hgetall(keyName)
keyTTL=redis.ttl(keyName)

logging.info("Key {} was set at {} and has {} seconds until expired".format(keyName,
    keyValues, keyTTL))
```

Per eseguire il programma, immetti il comando seguente:

```
python SetAndGetHash.py
```

Publicare (scrivere) e iscriversi (leggere) da un canale Pub/Sub

Copia il seguente programma e incollalo in un file denominato PubAndSub.py.

```
import logging
import time

def handlerFunction(message):
    """Prints message got from PubSub channel to the log output

    Return None
    :param message: message to log
    """
    logging.info(message)

logging.basicConfig(level=logging.INFO)
redis = Redis(host="redis202104053.tihewd.ng.0001.use1.cache.amazonaws.com", port=6379,
    decode_responses=True)

# Creates the subscriber connection on "mychannel"
subscriber = redis.pubsub()
subscriber.subscribe(**{'mychannel': handlerFunction})

# Creates a new thread to watch for messages while the main process continues with its
routines
thread = subscriber.run_in_thread(sleep_time=0.01)

# Creates publisher connection on "mychannel"
redis.publish('mychannel', 'My message')

# Publishes several messages. Subscriber thread will read and print on log.
```

```
while True:
    redis.publish('mychannel',time.ctime(time.time()))
    time.sleep(1)
```

Per eseguire il programma, immetti il comando seguente:

```
python PubAndSub.py
```

Scrivere e leggere da un flusso

Copia il seguente programma e incollalo in un file denominato ReadWriteStream.py.

```
from redis import Redis
import redis.exceptions as exceptions
import logging
import time
import threading

logging.basicConfig(level=logging.INFO)

def writeMessage(streamName):
    """Starts a loop writting the current time and thread name to 'streamName'

    :param streamName: Stream (key) name to write messages.
    """
    fieldsDict={'writerId':threading.currentThread().getName(),'myvalue':None}
    while True:
        fieldsDict['myvalue'] = time.ctime(time.time())
        redis.xadd(streamName,fieldsDict)
        time.sleep(1)

def readMessage(groupName=None,streamName=None):
    """Starts a loop reading from 'streamName'
    Multiple threads will read from the same stream consumer group. Consumer group is
    used to coordinate data distribution.
    Once a thread acknowleges the message, it won't be provided again. If message
    wasn't acknowledged, it can be served to another thread.

    :param groupName: stream group were multiple threads will read.
    :param streamName: Stream (key) name where messages will be read.
    """

    readerID=threading.currentThread().getName()
    while True:
```

```
    try:
        # Check if the stream has any message
        if redis.xlen(streamName)>0:
            # Check if if the messages are new (not acknowledged) or not (already
            processed)
            streamData=redis.xreadgroup(groupName,readerID,
{streamName:'>'},count=1)
            if len(streamData) > 0:
                msgId,message = streamData[0][1][0]
                logging.info("{}: Got {} from ID
{}".format(readerID,message,msgId))
                #Do some processing here. If the message has been processed
                sucessfully, acknowledge it and (optional) delete the message.
                redis.xack(streamName,groupName,msgId)
                logging.info("Stream message ID {} read and processed successfully
                by {}".format(msgId,readerID))
                redis.xdel(streamName,msgId)
            else:
                pass
        except:
            raise

        time.sleep(0.5)

# Creates the stream 'mystream' and consumer group 'myworkergroup' where multiple
threads will write/read.
try:
    redis.xgroup_create('mystream','myworkergroup',mkstream=True)
except exceptions.ResponseError as e:
    logging.info("Consumer group already exists. Will continue despite the error:
{}".format(e))
except:
    raise

# Starts 5 writer threads.
for writer_no in range(5):
    writerThread = threading.Thread(target=writeMessage, name='writer-'+str(writer_no),
args=('mystream',),daemon=True)
    writerThread.start()

# Starts 10 reader threads
for reader_no in range(10):
    readerThread = threading.Thread(target=readMessage, name='reader-'+str(reader_no),
args=('myworkergroup','mystream',),daemon=True)
```

```
readerThread.daemon = True
readerThread.start()

# Keep the code running for 30 seconds
time.sleep(30)
```

Per eseguire il programma, immetti il comando seguente:

```
python ReadWriteStream.py
```

Tutorial: Configurazione di una funzione Lambda per accedere ad Amazon in un ElastiCache Amazon VPC

In questo tutorial puoi imparare a creare una cache ElastiCache serverless, creare una funzione Lambda, quindi testare la funzione Lambda e, facoltativamente, ripulirla dopo.

Argomenti

- [Fase 1: Creare una cache serverless](#)
- [Passaggio 2: creazione di una funzione Lambda](#)
- [Fase 3: esecuzione del test della funzione Lambda](#)
- [Fase 4: Pulizia \(opzionale\)](#)

Fase 1: Creare una cache serverless

Per creare una cache serverless, segui questi passaggi.

Argomenti

- [Passaggio 1.1: Creare una cache serverless](#)
- [Passaggio 1.2: Copia l'endpoint della cache serverless](#)
- [Fase 1.3: Creazione del ruolo IAM](#)
- [Fase 1.4: Creare una cache serverless](#)

Passaggio 1.1: Creare una cache serverless

In questo passaggio, crei una cache serverless nel VPC Amazon predefinito nella regione us-east-1 del tuo account utilizzando (CLI AWS Command Line Interface). Per informazioni sulla creazione

di cache serverless utilizzando la console o l'API, consulta ElastiCache . [Fase 1: creazione di una cache](#)

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name cache-01 \  
  --description "ElastiCache IAM auth application" \  
  --engine redis
```

Il valore del campo Stato è impostato su CREATING. Il completamento della creazione della cache può richiedere un minuto. ElastiCache

Passaggio 1.2: Copia l'endpoint della cache serverless

Verifica che ElastiCache for Redis abbia terminato la creazione della cache con il comando. `describe-serverless-caches`

```
aws elasticache describe-serverless-caches \  
  --serverless-cache-name cache-01
```

Copia l'indirizzo dell'endpoint presente 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`.

```
{  
  "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`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect" : "Allow",
      "Action" : [
        "elasticache:Connect"
      ],
      "Resource" : [
        "arn:aws:elasticache:us-east-1:123456789012:serverlesscache:cache-01",
        "arn:aws:elasticache:us-east-1:123456789012:user:iam-user-01"
      ]
    }
  ]
}
```

3. Crea un ruolo IAM.

```
aws iam create-role \
--role-name "elasticache-iam-auth-app" \
--assume-role-policy-document file://trust-policy.json
```

4. Creare la policy IAM.

```
aws iam create-policy \
--policy-name "elasticache-allow-all" \
--policy-document file://policy.json
```

5. Allegare la policy IAM al ruolo.

```
aws iam attach-role-policy \
--role-name "elasticache-iam-auth-app" \
--policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"
```

Fase 1.4: Creare una cache serverless

1. Crea un nuovo utente predefinito.

```
aws elasticache create-user \  
  --user-name default \  
  --user-id default-user-disabled \  
  --engine redis \  
  --authentication-mode Type=no-password-required \  
  --access-string "off +get ~keys"
```

2. Crea un nuovo utente attivato da IAM.

```
aws elasticache create-user \  
  --user-name iam-user-01 \  
  --user-id iam-user-01 \  
  --authentication-mode Type=iam \  
  --engine redis \  
  --access-string "on ~* +@all"
```

3. Crea un gruppo di utenti e collega l'utente.

```
aws elasticache create-user-group \  
  --user-group-id iam-user-group-01 \  
  --engine redis \  
  --user-ids default-user-disabled iam-user-01  
  
aws elasticache modify-serverless-cache \  
  --serverless-cache-name cache-01 \  
  --user-group-id iam-user-group-01
```

Passaggio 2: creazione di una funzione Lambda

Per creare una funzione Lambda, procedi nel seguente modo.

Argomenti

- [Fase 2.1: Creare una funzione Lambda](#)
- [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: Creare una funzione Lambda

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 nella cache. ElastiCache Copia il codice e salvalo in un file denominato `app.py`. Assicurati di sostituire il `elasticache_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 ElastiCacheIAMProvider(redis.CredentialProvider):
    def __init__(self, user, cache_name, is_serverless=False, region="us-east-1"):
        self.user = user
        self.cache_name = cache_name
        self.is_serverless = is_serverless
        self.region = region

        session = boto3.session.get_session()
        self.request_signer = RequestSigner(
            ServiceId("elasticache"),
            self.region,
            "elasticache",
            "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}
            if self.is_serverless:
                query_params["ResourceType"] = "ServerlessCache"
```

```

    url = urlparse(
        ParseResult(
            scheme="https",
            netloc=self.cache_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
    # Elasticache 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
    cache_name = "cache-01" # replace with your cache name
    elasticache_endpoint = "cache-01-xxxxx.serverless.us1.cache.amazonaws.com" #
    replace with your cache endpoint
    creds_provider = ElastiCacheIAMProvider(user=username, cache_name=cache_name,
    is_serverless=True)
    redis_client = redis.Redis(host=elasticache_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 cache
    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 cache and print
    # the results
    if decoded_result == uuid_in:
        print(f"Success: Inserted {uuid_in}. Fetched {decoded_result} from Redis.")
    else:
        raise Exception(f"Bad value retrieved. Expected {uuid_in}, got
        {decoded_result}")

```

```
return "Fetched value from Redis"
```

Questo codice utilizza la libreria Python redis-py per inserire elementi nella cache e recuperarli. Questo codice utilizza cachetools per memorizzare nella cache i token IAM Auth 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 di codice sorgente app.py, create un pacchetto di cartelle in cui installare le librerie redis-py e cachetools.

```
mkdir package
```

Installa redis-py, cachetools usando pip.

```
pip install --target ./package redis
pip install --target ./package cachetools
```

Crea un file.zip contenente le librerie redis-py e cachetools. Su Linux e macOS, esegui il comando seguente. In Windows, utilizzate l'utilità zip preferita per creare un file.zip con le librerie redis-py e cachetools 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 .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 AWSLambdaVPCAccessExecutionRole al ruolo.

```
aws iam attach-role-policy \
  --role-name "elasticache-iam-auth-app" \
  --policy-arn "arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole"
```

Fase 2.3: caricamento del pacchetto di implementazione (creazione della funzione Lambda)

In questo passaggio, crei la funzione Lambda (AccessRedis) 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 AccessRedis \  
--region us-east-1 \  
--zip-file fileb://my_deployment_package.zip \  
--role arn:aws:iam::123456789012:role/elasticache-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 `AccessRedis` () utilizzando AWS Lambda il comando `invoke`.

```
aws lambda invoke \  
--function-name AccessRedis \  
--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 Redis.
```

- 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 la cache serverless](#)
- [Fase 4.3: Rimuovere il ruolo e le politiche IAM](#)

Fase 4.1: Eliminare la funzione Lambda

```
aws lambda delete-function \  
--function-name AccessRedis
```

Fase 4.2: Eliminare la cache serverless

Eliminare la cache.

```
aws elasticache delete-serverless-cache \  
--serverless-cache-name cache-01
```

Rimuovi utenti e gruppi di utenti.

```
aws elasticache delete-user \  

```



```
--user-id default-user-disabled

aws elasticache delete-user \
  --user-id iam-user-01

aws elasticache delete-user-group \
  --user-group-id iam-user-group-01
```

Fase 4.3: Rimuovere il ruolo e le politiche IAM

```
aws iam detach-role-policy \
  --role-name "elasticache-iam-auth-app" \
  --policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"

aws iam detach-role-policy \
  --role-name "elasticache-iam-auth-app" \
  --policy-arn "arn:aws:iam::aws:policy/service-role/AWSLambdaVPCAccessExecutionRole"

aws iam delete-role \
  --role-name "elasticache-iam-auth-app"

aws iam delete-policy \
  --policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"
```

Progettazione e gestione del proprio cluster ElastiCache

Se hai bisogno di un controllo granulare sul cluster ElastiCache, puoi scegliere di progettare il tuo cluster. ElastiCache consente di usare un cluster basato su nodi, scegliendo il tipo di nodo, il numero di nodi e il posizionamento dei nodi nelle zone di disponibilità AWS del cluster. Poiché ElastiCache è un servizio completamente gestito, automaticamente fornisce il provisioning dell'hardware, il monitoraggio, la sostituzione dei nodi e l'applicazione delle patch al software del cluster.

Per informazioni sulla configurazione, consulta [Configurazione](#). Per i dettagli sulla gestione, l'aggiornamento o l'eliminazione di nodi o cluster, consulta [Gestione di nodi](#). Per una panoramica dei componenti principali di un'implementazione Amazon ElastiCache quando si progetta autonomamente un cluster ElastiCache, consulta questi [concetti chiave](#).

Argomenti

- [ElastiCache per componenti e funzionalità Redis](#)
- [Terminologia di ElastiCache per Redis](#)
- [Progettazione del proprio cluster](#)
- [Gestione di nodi](#)
- [Gestione dei cluster](#)
- [Confronto tra cache progettate autonomamente Memcached e Redis](#)
- [Migrazione online a ElastiCache](#)
- [Scelta di regioni e zone di disponibilità](#)

ElastiCache per componenti e funzionalità Redis

Di seguito, puoi trovare una panoramica dei componenti principali di una ElastiCache distribuzione Amazon.

Argomenti

- [ElastiCache nodi](#)
- [ElastiCache per gli shard Redis](#)
- [ElastiCache per i cluster Redis](#)
- [ElastiCache per la replica Redis](#)
- [AWS Regioni e zone di disponibilità](#)

- [ElastiCache per gli endpoint Redis](#)
- [ElastiCache gruppi di parametri](#)
- [ElastiCache per la sicurezza Redis](#)
- [ElastiCache gruppi di sottoreti](#)
- [ElastiCache per i backup Redis](#)
- [ElastiCache eventi](#)

ElastiCache nodi

Un nodo è l'elemento costitutivo più piccolo di una ElastiCache distribuzione. Un nodo può essere isolato o avere qualche tipo di rapporto con altri nodi.

Un nodo è un blocco di dimensioni fisse di RAM collegata alla rete. Ogni nodo esegue un'istanza del motore e della versione scelti 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 di istanza diverso. Per ulteriori informazioni, consulta [Scalabilità per Redis ElastiCache](#).

Ogni nodo all'interno di un cluster è dello stesso tipo di istanza ed esegue lo stesso motore di cache. Ogni nodo di cache dispone del proprio Domain Name Service (DNS) e della propria porta. Sono supportati più tipi di nodi di cache, ciascuno con quantità variabili di memoria associata. Per una lista di tipi di istanze di nodo supportate, consulta [Tipi di nodi supportati](#).

Puoi acquistare nodi su pay-as-you-go base volontaria, pagando solo per l'utilizzo di un nodo. Oppure puoi acquistare nodi riservati a una tariffa oraria significativamente ridotta. Se il tasso di utilizzo è elevato, l'acquisto di nodi riservati aiuta a risparmiare. Supponiamo che il cluster sia quasi sempre in uso e che di tanto in tanto vengano aggiunti nodi per gestire i picchi di utilizzo. In tal caso, è possibile acquistare un numero di nodi riservati che verranno eseguiti nella maggior parte del tempo. È quindi possibile acquistare pay-as-you-go nodi per le volte in cui è necessario aggiungere nodi. Per ulteriori informazioni sui nodi riservati, consulta [Nodi riservati ElastiCache](#).

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

ElastiCache per gli shard Redis

Una partizione di Redis (chiamato gruppo di nodi nell'API e nella CLI) è un raggruppamento composto da uno a sei nodi correlati. Un cluster Redis (modalità cluster disabilitata) ha sempre almeno uno shard.

Lo sharding è un metodo di partizionamento del database che separa i database di grandi dimensioni in parti più piccole, più veloci e più facili da gestire, chiamate frammenti di dati. Ciò può aumentare l'efficienza del database distribuendo le operazioni su più sezioni separate. L'utilizzo degli shard può offrire molti vantaggi, tra cui miglioramento delle prestazioni, della scalabilità e dell'efficienza dei costi.

I cluster di Redis (modalità cluster abilitata) possono avere fino a 500 partizioni, con i dati partizionati tra le partizioni. Il limite del nodo o della partizione può essere aumentato fino a un massimo di 500 per cluster se la versione del motore Redis è 5.0.6 o superiore. Ad esempio, è possibile scegliere di configurare un cluster a 500 nodi che varia tra 83 partizioni (un primario e 5 repliche per partizione) e 500 partizioni (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. Per ulteriori informazioni, consulta [Creazione di un gruppo di sottoreti](#). Per le versioni inferiori alla 5.0.6, il limite è 250 per cluster.

Per richiedere un aumento dei limiti, consulta [Limiti dei servizi AWS](#) e seleziona il tipo di limite Nodi per cluster per tipo di istanza.

Una partizione a nodo multiplo implementa repliche tramite un nodo primario di lettura/scrittura e 1-5 nodi di replica. Per ulteriori informazioni, consulta [Alta disponibilità utilizzando gruppi di replica](#).

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

ElastiCache per i cluster Redis

Un cluster Redis è un raggruppamento logico di uno o più [ElastiCache per gli shard Redis](#). I dati vengono partizionati tra le partizioni in un cluster di Redis (modalità cluster abilitata).

Molte ElastiCache operazioni sono rivolte ai cluster:

- Creazione di un cluster
- Modifica di un cluster
- Esecuzione di snapshot di un cluster (tutte le versioni di Redis)
- 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.

- [AWS limiti di servizio: Amazon ElastiCache](#)

Informazioni sui ElastiCache limiti, come il numero massimo di nodi o cluster. Per superare alcuni di questi limiti, puoi effettuare una richiesta utilizzando il [modulo di richiesta del nodo di ElastiCache cache Amazon](#).

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

Informazioni su come migliorare la tolleranza ai guasti dei cluster e dei gruppi di replica.

Configurazioni tipiche dei cluster

Di seguito sono riportate le configurazioni tipiche dei cluster.

Cluster di Redis

I cluster di Red (modalità cluster disabilitata) contengono sempre una sola partizione (nell'API e nella CLI, un gruppo di nodi). Una partizione di Redis contiene da uno a sei nodi. Se una partizione contiene più nodi, la partizione supporta la funzione di replica. In tal caso, un nodo è il nodo primario di lettura/scrittura e gli altri sono nodi di replica di sola lettura.

Per una migliore tolleranza ai guasti, ti consigliamo di avere almeno due nodi in un cluster Redis e di abilitare la funzione Multi-AZ. Per ulteriori informazioni, consulta [Limitazione dell'impatto degli errori](#).

Quando la domanda relativa al cluster di Redis (modalità cluster disabilitata) cambia, puoi eseguire il dimensionamento verso l'alto o verso il basso. A tale scopo, sposta il cluster in un tipo di istanza del nodo diverso. Se l'applicazione è di lettura intensiva, ti consigliamo di aggiungere cluster di Redis (modalità cluster disabilitata) di repliche di sola lettura. In tal modo, puoi distribuire le operazioni di lettura su un numero più adatto di nodi.

Puoi anche utilizzare il tiering di dati. I dati a cui si accede con maggiore frequenza vengono archiviati su memoria e i dati a cui si accede con minore frequenza vengono archiviati su disco. L'utilizzo del tiering di dati ha il vantaggio di ridurre la memoria necessaria. Per ulteriori informazioni, consulta [Tiering di dati](#).

ElastiCache supporta la modifica dinamica del tipo di nodo di un cluster Redis (modalità cluster disabilitata) in un tipo di nodo più grande. Per informazioni sul dimensionamento verso l'alto o

verso il basso, consulta [Ridimensionamento dei cluster a nodo singolo per Redis \(modalità cluster disabilitata\)](#) o [Ridimensionamento dei cluster Redis \(modalità cluster disabilitata\) con nodi di replica](#).

ElastiCache per la replica Redis

La funzione di replica viene implementata tramite il raggruppamento da due a sei nodi in una partizione (denominato gruppo di nodi nell'API e nella CLI). Uno di questi nodi è quello di primario di lettura/scrittura. Tutti gli altri sono nodi di replica di sola lettura.

Ogni nodo di replica conserva una copia dei dati del nodo primario. I nodi di replica utilizzano meccanismi di replica asincrona per mantenere la sincronizzazione con il nodo primario. Le applicazioni possono eseguire la lettura da ogni nodo del cluster, ma possono scrivere solo sui nodi primari. Le repliche di lettura migliorano la scalabilità distribuendo le operazioni di lettura in più endpoint. Le repliche di lettura migliorano anche la tolleranza ai guasti conservando più copie dei dati. La tolleranza ai guasti migliora ulteriormente se si posizionano le repliche di lettura in più zone di disponibilità. Per ulteriori informazioni sulla tolleranza ai guasti, consulta [Limitazione dell'impatto degli errori](#).

I cluster di Redis (modalità cluster disabilitata) supportano una partizione (nell'API e nella CLI, chiamata gruppo di nodi).

La funzione di replica nell'API e nell'CLI utilizza una terminologia diversa per mantenere la compatibilità con le versioni precedenti, ma i risultati sono gli stessi. La seguente tabella mostra i termini dell'API e dell'CLI per l'implementazione della funzione di replica.

Confronto di replica: Redis (modalità cluster disabilitata) e Redis (modalità cluster abilitata)

Nella tabella seguente è possibile trovare un confronto tra le funzionalità dei gruppi di replica di Redis (modalità cluster disabilitata) e Redis (modalità cluster abilitata).

	Redis (modalità cluster disabilitata)	Redis (modalità cluster abilitata)
Partizioni (gruppi di nodi)	1	1–500
Repliche per ogni partizione (gruppo di nodi)	0–5	0–5
Partizionamento dei dati	No	Sì

	Redis (modalità cluster disabilitata)	Redis (modalità cluster abilitata)
Aggiungere/eliminare repliche	Sì	Sì
Aggiungere/eliminare gruppi di nodi	No	Sì
Supporta il dimensionamento	Sì	Sì
Supporta gli aggiornamenti del motore	Sì	Sì
Promuovere la replica al primario	Sì	Automatica
Multi-AZ	Facoltativo	Richiesto
Ripristino backup	Sì	Sì

Note:

Se un primario non dispone di repliche e il primario ha esito negativo, si perdono tutti i dati di tale primario.

È possibile utilizzare il backup e il ripristino per eseguire la migrazione a Redis (modalità cluster abilitata).

È possibile utilizzare il backup e il ripristino per ridimensionare il cluster di Redis (modalità cluster abilitata).

Tutti le partizioni (nell'API e nella CLI, gruppi di nodi) e i nodi devono risiedere nella stessa regione AWS . Tuttavia, è possibile effettuare il provisioning dei singoli nodi in più zone di disponibilità all'interno di quella AWS regione.

Le repliche di lettura proteggono da potenziali perdite di dati perché i dati vengono replicati su due o più nodi, ovvero la primaria e una o più repliche di lettura. Per ottenere una maggiore affidabilità e un recupero più rapido, consigliamo di creare una o più repliche di lettura in differenti zone di disponibilità.

È inoltre possibile utilizzare Global datastore. Utilizzando la funzionalità Global Datastore for Redis, puoi lavorare con una replica completamente gestita, veloce, affidabile e sicura in tutte le regioni. AWS Utilizzando questa funzionalità, è possibile creare cluster di replica di lettura interregionali per Redis ElastiCache per consentire letture a bassa latenza e disaster recovery tra le regioni. AWS [Per ulteriori informazioni, consulta Replica tra regioni utilizzando datastore globali. AWS](#)

Funzione di replica: limiti ed esclusioni

- La funzione Multi-AZ non è supportata sui tipi di nodo T1.

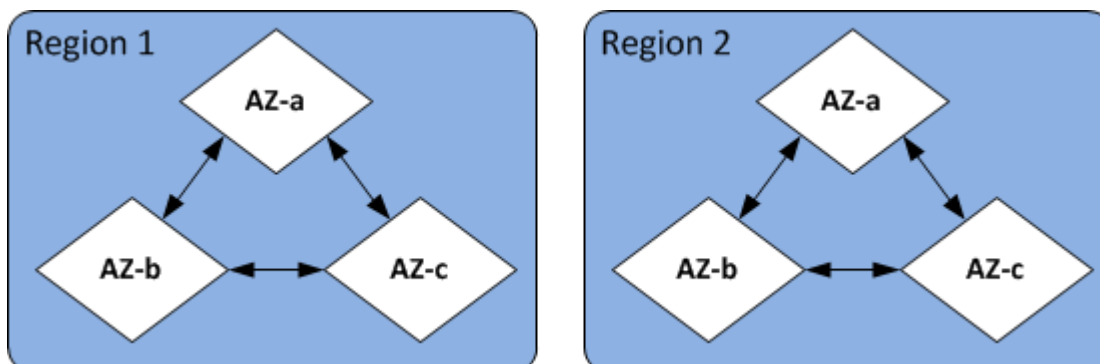
AWS Regioni e zone di disponibilità

Amazon ElastiCache è disponibile in diverse AWS regioni in tutto il mondo. In questo modo, puoi avviare ElastiCache cluster nelle località che soddisfano i tuoi requisiti aziendali. Ad esempio, puoi lanciare nella AWS regione più vicina ai tuoi clienti o per soddisfare determinati requisiti legali.

Per impostazione predefinita, gli AWS SDK AWS CLI, l'ElastiCache API e la ElastiCache console fanno riferimento alla regione Stati Uniti occidentali (Oregon). Man mano ElastiCache che la disponibilità viene estesa a nuove AWS regioni, sono disponibili anche nuovi endpoint per queste AWS regioni. Puoi usarli nelle tue richieste HTTP, negli AWS SDK e nella console AWS CLI.

ElastiCache

Ogni AWS regione è progettata per essere completamente isolata dalle altre AWS regioni. All'interno di ciascuna regione sono presenti più zone di disponibilità. Avviando i nodi in diverse zone di disponibilità, puoi ottenere la massima tolleranza ai guasti possibile. Per ulteriori informazioni su AWS regioni e zone di disponibilità, vedere [Scelta di regioni e zone di disponibilità](#). Nel diagramma seguente, è possibile visualizzare una panoramica di alto livello del funzionamento delle AWS regioni e delle zone di disponibilità.



Per informazioni sulle AWS regioni supportate da ElastiCache e sui relativi endpoint, consulta [Regioni ed endpoint supportati](#).

ElastiCache per gli endpoint Redis

Un endpoint è l'indirizzo univoco utilizzato dall'applicazione per connettersi a un ElastiCache nodo o cluster.

Endpoint a nodo singolo per Redis (modalità cluster disabilitata)

L'endpoint di un cluster Redis a nodo singolo è utilizzato per la connessione al cluster per le operazioni di lettura e di scrittura.

Endpoint multi-nodo per Redis (modalità cluster disabilitata)

Un cluster Redis a più nodi (modalità cluster disabilitata) dispone di due tipi di endpoint. L'endpoint primario si connette sempre al nodo primario del cluster, anche se cambia il nodo specifico nel ruolo primario. Utilizza l'endpoint primario per tutte le operazioni di scrittura sul cluster.

Usa l'endpoint di lettura per dividere equamente le connessioni in arrivo verso l'endpoint tra tutte le repliche di lettura. I vari endpoint di nodo (definiti in ambiente API/CLI endpoint di lettura) vengono invece impiegati per le operazioni di lettura.

Endpoint di Redis (modalità cluster abilitata)

Un cluster Redis (modalità cluster abilitata) dispone di un solo endpoint di configurazione. Connettendosi all'endpoint di configurazione, la tua applicazione è in grado di scoprire gli endpoint primari e di lettura per ogni partizione del cluster.

Per ulteriori informazioni, consulta [Individuazione degli endpoint di connessione](#).

ElastiCache gruppi di parametri

I gruppi di parametri cache sono un modo facile per gestire le impostazioni di runtime per il software di motore supportato. I parametri vengono utilizzati per controllare l'utilizzo della memoria, le policy di espulsione, le dimensioni degli elementi e altro ancora. Un gruppo di ElastiCache parametri è una raccolta denominata di parametri specifici del motore che è possibile applicare a un cluster. In questo modo, sei certo che tutti i nodi di quel cluster siano configurati esattamente nello stesso modo.

Per un elenco dei parametri supportati, dei valori predefiniti e di quelli che si possono modificare, consulta [DescribeEngineDefaultParameters](#) (CLI: [describe-engine-default-parameters](#)).

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

ElastiCache per la sicurezza Redis

Per una maggiore sicurezza, ElastiCache per Redis l'accesso ai nodi è limitato alle applicazioni in esecuzione sulle istanze Amazon EC2 consentite. Puoi controllare le istanze Amazon EC2 che possono accedere al cluster utilizzando i gruppi di sicurezza.

Per impostazione predefinita, tutte le novità ElastiCache per i cluster Redis vengono lanciate in un ambiente Amazon Virtual Private Cloud (Amazon VPC). Puoi utilizzare i gruppi di sottoreti per concedere l'accesso al cluster dalle istanze Amazon EC2 in esecuzione su sottoreti specifiche.

Oltre a limitare l'accesso ai nodi, ElastiCache for Redis supporta TLS e la crittografia sul posto per i nodi che eseguono versioni specifiche di for Redis. ElastiCache Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Sicurezza dei dati in Amazon ElastiCache](#)
- [Autenticazione con il comando Redis AUTH](#)

ElastiCache 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 VPC.

Se crei un cluster in un Amazon VPC, devi specificare un gruppo di sottoreti della cache. ElastiCache utilizza quel gruppo di sottoreti di cache per scegliere una sottorete e gli indirizzi IP all'interno di quella sottorete da associare ai nodi di cache.

Per ulteriori informazioni sull'utilizzo dei gruppi di sottoreti della cache in un ambiente Amazon VPC, consulta quanto segue:

- [Amazon VPC e sicurezza di ElastiCache](#)
- [Fase 3: autorizzazione dell'accesso al cluster](#)
- [Sottoreti e gruppi di sottoreti](#)

ElastiCache per i backup Redis

Un backup è una point-in-time copia di un cluster Redis. I backup possono essere utilizzati per ripristinare un cluster esistente o per inizializzarne uno nuovo. I backup sono costituiti da tutti i dati di un cluster e da alcuni metadati.

A seconda della versione di Redis in esecuzione sul cluster, il processo di backup richiede quantità diverse di memoria riservata per andare a buon fine. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Snapshot e ripristino](#)
- [Modalità di implementazione di sincronizzazione e backup](#)
- [Impatto sulle prestazioni dei backup di cluster progettati autonomamente](#)
- [Verifica della disponibilità di memoria sufficiente per creare uno snapshot Redis](#)

ElastiCache eventi

Quando si verificano eventi importanti in un cluster di cache, ElastiCache invia una notifica a un argomento specifico di Amazon SNS. Questi eventi possono includere, ad esempio, l'impossibilità o la possibilità di aggiungere un nodo,, la modifica di un gruppo di sicurezza e altro ancora. Monitorando gli eventi chiave, puoi conoscere lo stato corrente dei cluster e in molti casi intraprendere operazioni correttive.

Per ulteriori informazioni sugli ElastiCache eventi, consulta [Monitoraggio di Amazon SNS per gli eventi ElastiCache](#).

Terminologia di ElastiCache per Redis

Nell'ottobre 2016, Amazon ElastiCache ha lanciato il supporto per Redis 3.2. A quel punto, abbiamo aggiunto il supporto per il partizionamento dei dati su un massimo di 500 partizioni (chiamati gruppi di nodi nell'API di ElastiCache e nella AWS CLI). Per conservare la compatibilità con le versioni precedenti, abbiamo ampliato le operazioni della versione 2015-02-02 dell'API per includere la nuova funzionalità Redis.

Allo stesso tempo abbiamo iniziato a utilizzare nella console di ElastiCache la terminologia della nuova funzionalità, comune in tutto il settore. Queste modifiche implicano che in qualche caso la terminologia utilizzata nell'API e nell'CLI può essere diversa da quella utilizzata nella console. Il seguente elenco identifica i termini che potrebbero essere diversi dall'API e dalla CLI rispetto alla console.

Cluster o nodo di cache e nodo

Esiste una relazione uno-a-uno tra un nodo e un cluster di cache quando non ci sono nodi di replica. Pertanto, la console di ElastiCache utilizzava spesso i termini in modo interscambiabile. Ora la console utilizza il termine nodo ovunque. L'unica eccezione è il pulsante Create Cluster (Crea cluster), che avvia il processo di creazione di un cluster, con o senza nodi di replica.

L'API di ElastiCache e la AWS CLI continuano a utilizzare i termini nel modo utilizzato in precedenza.

Cluster e gruppo di replica

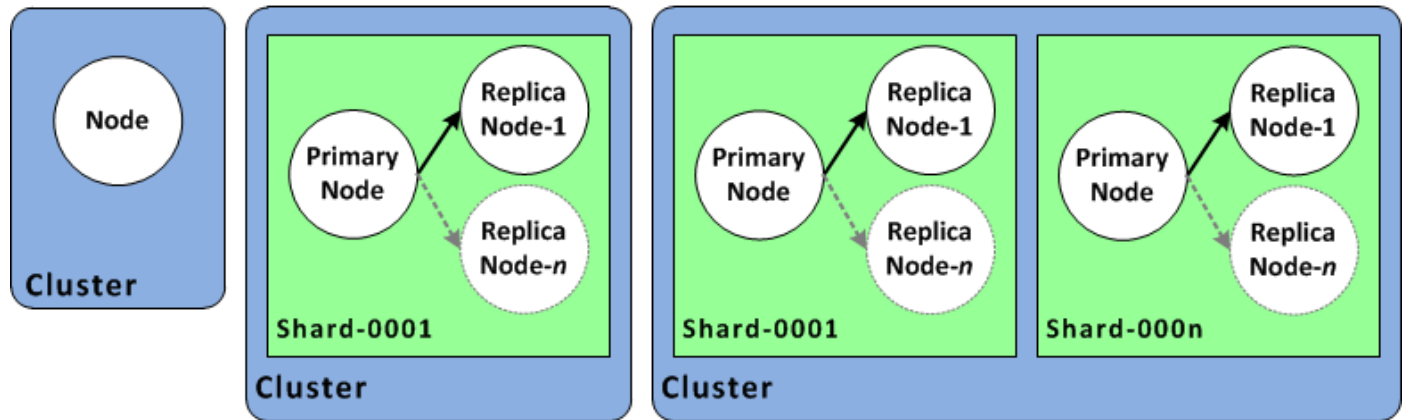
La console ora utilizza il termine cluster per tutti i cluster di ElastiCache per Redis. La console utilizza il termine cluster in tutti questi casi:

- Quando il cluster è un cluster Redis a nodo singolo.
- Quando il cluster è un cluster di Redis (modalità cluster disabilitata) che supporta la replica all'interno di una sola partizione (nell'API e nella CLI denominato gruppo di nodi).
- Quando il cluster è un cluster di Redis (modalità cluster abilitata) che supporta la replica all'interno di 1-90 partizioni o fino a 500 con una richiesta di aumento dei limiti. Per richiedere un aumento dei limiti, consulta [limiti del servizio AWS](#) e seleziona il tipo di limite Nodi per cluster per tipo di istanza.

Per ulteriori informazioni sui gruppi di replica, consulta [Alta disponibilità utilizzando gruppi di replica](#).

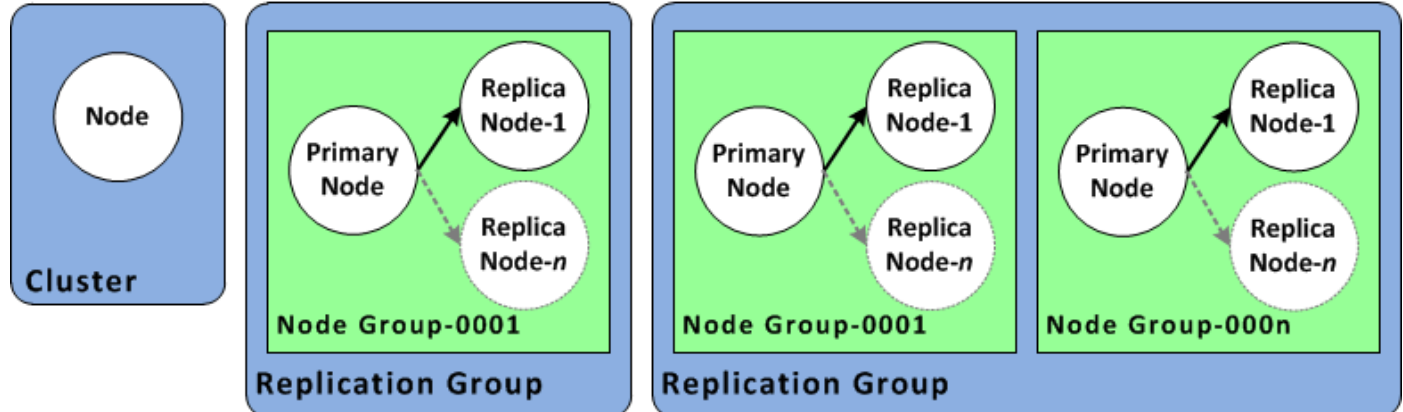
Il seguente diagramma illustra le varie topologie di cluster di ElastiCache per Redis dal punto di vista della console.

ElastiCache for Redis: Console View



Le operazioni dell'API di ElastiCache e della AWS CLI fanno ancora distinzione tra i cluster a nodo singolo di ElastiCache per Redis e i gruppi di replica multinodo. Il seguente diagramma illustra le varie topologie di ElastiCache per Redis dal punto di vista dell'API di ElastiCache e della AWS CLI.

ElastiCache for Redis: API/CLI View



Gruppo di replica e datastore globale

Un datastore globale è una raccolta di uno o più cluster che si replicano tra loro nelle regioni, mentre un gruppo di replica replica i dati in un cluster abilitato per la modalità cluster con più partizioni. Un Global Datastore è costituito dai seguenti elementi:

- **Cluster primario (attivo)**— Un cluster primario accetta scritture replicate in tutti i cluster all'interno del datastore globale. Un cluster primario accetta anche le richieste di lettura.

- **Cluster secondario (passivo)**— Un cluster secondario accetta solo richieste di lettura e replica gli aggiornamenti dei dati da un cluster primario. Un cluster secondario deve trovarsi in una regione AWS diversa rispetto al cluster primario.

Per informazioni sui datastore globali, vedere [Replica tra regioni AWS mediante Global Datastore](#).

Progettazione del proprio cluster

Di seguito sono riportate le azioni una tantum da intraprendere per iniziare a progettare il ElastiCache cluster.

Argomenti

- [Configurazione](#)
- [Fase 1: creare un gruppo di sottoreti](#)
- [Fase 2: creazione di un cluster](#)
- [Fase 3: autorizzazione dell'accesso al cluster](#)
- [Fase 4: connessione a un nodo del cluster](#)
- [Passaggio 5: Eliminazione di un cluster](#)
- [Esercitazioni e video relativi a ElastiCache](#)
- [A questo punto come si può procedere?](#)

Configurazione

Prima di creare il cluster, assicurati di creare un gruppo di sottoreti. Un gruppo di sottorete della cache è una raccolta di sottoreti che è possibile assegnare ai cluster di cache in un VPC. Quando si avvia il cluster di cache in un VPC, è necessario scegliere un gruppo di sottorete della cache. Quindi, ElastiCache utilizza il gruppo di sottoreti della cache per assegnare indirizzi IP all'interno di quella sottorete per ogni nodo di cache nel cluster.

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.

Per ulteriori informazioni sulla configurazione di ElastiCache, consulta [Configurazione](#).

Fase 1: creare un gruppo di sottoreti

Le procedure seguenti illustrano come creare un gruppo di sottoreti chiamato `mysubnetgroup` con la console e la AWS CLI.

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 gestione di AWS e aprire la console di ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nell'elenco di navigazione, scegliere Subnet Groups (Gruppi di sottoreti).
3. Scegliere Create Subnet Group (Crea gruppo di sottoreti).
4. Nella procedura guidata Create subnet group (Crea gruppo di sottoreti), eseguire le operazioni seguenti: Dopo aver selezionato tutte le impostazioni desiderate, scegliere Yes, Create (Crea).
 - a. Nella casella Name (Nome), digitare un nome per il gruppo di sottoreti.
 - 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.
 - d. Negli elenchi Availability Zone (Zone di disponibilità) e Subnet ID (ID sottoreti), selezionare la zona di disponibilità o [zona locale](#) e l'ID della sottorete privata, quindi scegliere Add (Aggiungi).

Subnet group settings

A subnet group is a collection of subnets (typically private). Designate a subnet group for your clusters running in an Amazon Virtual Private Cloud (VPC) environment.

Name

The name is required, can have up to 255 characters, and must begin with a letter. It should not end with a hyphen or contain two consecutive hyphens. Valid characters: A-Z, a-z, 0-9, and - (hyphen).

Description - optional

VPC ID
 The identifier for the VPC environment where your cluster is to run.
 [Create VPC](#)

For Multi-AZ high availability mode, choose IDs for at least two subnets from two Availability Zones in the table below.

Selected subnets (6) [Manage](#)

Availability Zone ▲	Subnet ID ▼	Outpost ID ▼	CIDR block ▼
us-east-1a	subnet- <input type="text"/>		172.31.16.0/20
us-east-1b	subnet- <input type="text"/>		172.31.32.0/20
us-east-1c	subnet- <input type="text"/>		172.31.0.0/20
us-east-1d	subnet- <input type="text"/>		172.31.80.0/20

5. Nel messaggio di conferma visualizzato, scegliere Close (Chiudi).

Il nuovo gruppo di sottoreti viene visualizzato nell'elenco Subnet Groups (Gruppi di sottoreti) della console di ElastiCache. 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.

Per creare un gruppo di sottoreti (AWS CLI)

Al prompt dei comandi, utilizzare il comando `create-cache-subnet-group` per creare un gruppo di sottoreti.

Per Linux, macOS o Unix:

```
aws elasticache create-cache-subnet-group \
  --cache-subnet-group-name mysubnetgroup \
  --cache-subnet-group-description "Testing" \
```

```
--subnet-ids subnet-53df9c3a
```

Per Windows:

```
aws elasticache create-cache-subnet-group ^  
  --cache-subnet-group-name mysubnetgroup ^  
  --cache-subnet-group-description "Testing" ^  
  --subnet-ids subnet-53df9c3a
```

Questo comando dovrebbe generare un output simile al seguente:

```
{  
  "CacheSubnetGroup": {  
    "VpcId": "vpc-37c3cd17",  
    "CacheSubnetGroupDescription": "Testing",  
    "Subnets": [  
      {  
        "SubnetIdentifier": "subnet-53df9c3a",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2a"  
        }  
      }  
    ],  
    "CacheSubnetGroupName": "mysubnetgroup"  
  }  
}
```

Per ulteriori informazioni, consulta l'argomento della AWS CLI [create-cache-subnet-group](#).

Fase 2: 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, verrà creato un cluster con la modalità cluster disabilitata e sarà possibile accettare i valori di configurazione predefiniti dove vengono applicati.

Il cluster che stai per avviare verrà eseguito in un ambiente attivo e non in una sandbox. Ti verranno addebitati i costi di ElastiCache utilizzo standard per l'istanza fino a quando non la elimini. 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 ElastiCache utilizzo, consulta [Amazon ElastiCache](#).

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

Creazione di un cluster Redis (modalità cluster disabilitata) (Console)


Per creare un cluster Redis (modalità cluster disabilitata) utilizzando la console ElastiCache

1. Accedi AWS Management Console e apri la ElastiCache console Amazon all'[indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Dall'elenco nell'angolo in alto a destra, scegli la AWS regione in cui desideri avviare questo cluster.
3. Scegli Nozioni di base dal riquadro di navigazione.
4. Scegli Crea VPC e segui la procedura descritta in [Creazione di un cloud privato virtuale \(VPC\)](#).
5. Nella pagina della ElastiCache dashboard, scegli Redis cache, quindi scegli Crea cache Redis.
6. In Cluster settings (Impostazioni cluster), procedi come indicato di seguito:
 - a. Scegli Configurazione e creazione di un nuovo cluster.
 - b. Per Modalità cluster, scegli Disabilitata.
 - c. Per Informazioni sul cluster, inserisci un valore per Nome.
 - d. (Facoltativo) Inserisci un valore per Descrizione.
7. In Location (Ubicazione):

AWS Cloud

1. Per CloudAWS , è preferibile accettare le impostazioni predefinite per Multi-AZ e Failover automatico. Per ulteriori informazioni, consulta [Ridurre al minimo i tempi di inattività in Redis with ElastiCache Multi-AZ](#).
2. In Cluster settings (Impostazioni cluster)
 - a. Per Engine version (Versione motore), scegli una versione disponibile.
 - b. Per Porta utilizza la porta predefinita, 6379. Se esiste un motivo per utilizzare una porta diversa, immettere il numero di porta.
 - c. Per Parameter group (Gruppo di parametri), scegliere un gruppo di parametri o crearne uno nuovo. I gruppi di parametri controllano i parametri di runtime del

cluster. Per ulteriori informazioni sui gruppi di parametri, consulta [Parametri specifici di Redis](#) e [Creazione di un gruppo di parametri](#).

 Note

Quando si seleziona un gruppo di parametri per impostare i valori di configurazione del motore, tale gruppo di parametri viene applicato a tutti i cluster nel Global Datastore. Nella pagina Parameter Groups (Gruppi di parametri) con/senza l'attributo Global (Globale) indica se un gruppo di parametri fa parte di un Global Datastore.

- d. In Node type (Tipo di nodo), scegliere la freccia in giù (▼)).
Nella finestra di dialogo Change node type (Cambia tipo di nodo) scegliere un valore per Instance family (Famiglia di istanze) per il tipo di nodo desiderato. Scegliere quindi il tipo di nodo che si desidera utilizzare per il cluster, quindi scegliere Save (Salva).

Per ulteriori informazioni, consulta [Scelta delle dimensioni dei nodi](#).

Se scegli un tipo di nodo r6gd, il tiering dei dati viene abilitato automaticamente. Per ulteriori informazioni, consulta [Tiering di dati](#).

- e. In Numero di repliche scegli il numero di repliche di lettura desiderato. Se hai abilitato Multi-AZ, il numero deve essere compreso tra 1 e 5.
3. In Connectivity (Connettività)
 - a. Per Network type (Tipo di rete), scegli le versioni IP supportate da questo cluster.
 - b. Per i gruppi di sottoreti, scegli la sottorete che desideri applicare a questo cluster. ElastiCache utilizza quel gruppo di sottorete per scegliere una sottorete e gli indirizzi IP all'interno di quella sottorete da associare ai nodi. ElastiCache i cluster richiedono una sottorete dual-stack con indirizzi IPv4 e IPv6 assegnati per funzionare in modalità dual-stack e una sottorete solo IPv6 per funzionare solo come IPv6.

Durante la creazione di un nuovo gruppo di sottoreti, immettere l'VPC ID (ID VPC) a cui appartiene.

Per ulteriori informazioni, consultare:

- [Scelta del tipo di rete.](#)
- Creazione di una sottorete nel VPC.

Se si esegue per [Utilizzo di local zones con ElastiCache](#), è necessario creare o scegliere una sottorete che si trova nella zona locale.

Per ulteriori informazioni, consulta [Sottoreti e gruppi di sottoreti.](#)

4. Per Posizionamento zone disponibilità, le opzioni sono due:

- ElastiCache Nessuna preferenza: sceglie la zona di disponibilità.
- Specify Availability Zones (Specifica zone di disponibilità) – L'utente specifica la zona di disponibilità per ogni cluster.


Se hai scelto di specificare le zone di disponibilità, per ogni cluster in ogni partizione, seleziona la zona di disponibilità dall'elenco.

Per ulteriori informazioni, consulta [Scelta di regioni e zone di disponibilità.](#)

5. Seleziona Successivo.

6. In Impostazioni avanzate di Redis

- Per Sicurezza:
 - i. 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 scegliendo la chiave AWS KMS gestita dal cliente e scegliendo la chiave. Per ulteriori informazioni, vedere [Utilizzo di chiavi gestite dal cliente da AWS KMS.](#)

- Crittografia dei dati in transito – Consente la crittografia dei dati in trasferimento. Per maggiori informazioni, consultare [Crittografia dei dati in transito.](#) Per il motore Redis versione 6.0 e successive, se si abilita la

Crittografia in transito, viene richiesto di specificare una delle seguenti opzioni di Controllo degli accessi:

- Nessun controllo degli accessi - Questa è l'impostazione di default. Questo non indica alcuna restrizione sull'accesso dell'utente al cluster.
- Lista di controllo accessi – scegliere un gruppo di utenti con un set definito di utenti che possono accedere al cluster. Per ulteriori informazioni, consulta [Gestione dei gruppi di utenti con Console e CLI](#).
- Redis AUTH Default User (Utente di default Redis AUTH) : Un meccanismo di autenticazione per il server Redis. Per ulteriori informazioni, vedere [Redis AUTH](#).
- Redis AUTH - Un meccanismo di autenticazione per il server Redis. Per ulteriori informazioni, vedere [Redis AUTH](#).

 Note

Per le versioni di Redis a partire da 3.2.6 e successive, esclusa la versione 3.2.10, Redis AUTH è l'unica opzione.

- ii. 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.

7. Se si desiderano backup regolari pianificati regolarmente, scegliere Enable automatic backups (Abilita backup automatici), quindi digitare il numero di giorni in cui conservare ciascun backup automatico prima che venga automaticamente eliminato. Se non si desiderano backup automatici pianificati regolarmente, deselezionare la casella di controllo Enable automatic backups (Abilita backup automatici). In ogni caso, hai sempre la possibilità di creare backup manuali.

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

8. (Facoltativo) Specifica una finestra di manutenzione. La Maintenance window (Finestra di manutenzione) è l'intervallo di tempo, generalmente un'ora a settimana, in cui ElastiCache pianifica la manutenzione di sistema per il cluster. Si può consentire a

ElastiCache di scegliere l'ora e il giorno della finestra di manutenzione (No preference (Nessuna preferenza)) o sceglierne autonomamente il giorno, l'ora e la durata (Specify maintenance window (Specifica 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](#).

9. (Facoltativo) PerRegistro:

- In Formato del registro, scegliere Text o JSON.
 - In Tipo di destinazione, scegli CloudWatch Logs o Kinesis Firehose.
 - In Destinazione log, scegliete Crea nuovo e inserite il nome del gruppo di log CloudWatch Logs o il nome dello stream Firehose, oppure scegliete Seleziona esistente e quindi scegliete il nome del gruppo di log Logs o CloudWatch il nome dello stream Firehose,
10. Per quanto riguarda i tag, per aiutarti a gestire i tuoi cluster e altre ElastiCache risorse, puoi assegnare i tuoi metadati a ciascuna risorsa sotto forma di tag. Per ulteriori informazioni, consulta [Assegnazione di tag delle risorse ElastiCache](#).
11. Seleziona Avanti.
12. Riesaminare le voci e le selezioni, quindi apportare le eventuali correzioni. Al termine, scegli Crea.

On premises

1. Per On-premise, è preferibile lasciare abilitato il Failover automatico. Per ulteriori informazioni, consulta [Ridurre al minimo i tempi](#) di inattività in Redis con Multi-AZ ElastiCache
2. Per completare la creazione del cluster, segui la procedura in [Utilizzo outpost](#).

Quando lo stato del cluster diventa disponibile, è possibile concedere a Amazon EC2 l'accesso al cluster, connettersi al cluster e iniziare a utilizzarlo. Per ulteriori informazioni, consultare [Fase 3: autorizzazione dell'accesso al cluster](#) e [Fase 4: connessione a un nodo del cluster](#).

⚠ Important

Quando il cluster è 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 [Eliminazione di un cluster](#).

Creazione di un cluster Redis (modalità cluster disabilitata) (AWS CLI)**Example**

Il seguente codice CLI crea un cluster di cache senza repliche.

Per Linux, macOS o Unix:

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--cache-node-type cache.r4.large \  
--engine redis \  
--num-cache-nodes 1 \  
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

Per Windows:

```
aws elasticache create-cache-cluster ^  
--cache-cluster-id my-cluster ^  
--cache-node-type cache.r4.large ^  
--engine redis ^  
--num-cache-nodes 1 ^  
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

Per utilizzare la modalità cluster abilitata, consulta i seguenti argomenti:

- Per utilizzare la console, consulta [Creazione di un cluster Redis \(modalità cluster abilitata\) \(console\)](#).
- Per utilizzare il, consulta. AWS CLI [Creazione di un cluster Redis \(modalità cluster abilitata\) \(AWS CLI\)](#)

Fase 3: autorizzazione dell'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 ElastiCache sono progettati per consentire l'accesso mediante un'istanza Amazon EC2. Lo scenario più comune è quello di accedere a un cluster ElastiCache da un'istanza Amazon EC2 nello stesso Amazon Virtual Private Cloud (Amazon VPC), che sarà il caso per questo esercizio.

Per impostazione predefinita, l'accesso di rete al cluster è riservato all'account che è stato utilizzato per avviarlo. Prima di poter eseguire la connessione a un cluster da un'istanza EC2, devi autorizzare l'istanza EC2 ad accedere al cluster. I passaggi richiesti dipendono dal fatto che il cluster sia stato avviato in EC2-VPC o EC2-Classic.

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 alle istanze del 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 Redis e i gruppi di replica è 6379.

Important

I gruppi di sicurezza di Amazon ElastiCache si applicano solo ai cluster che non sono in esecuzione in un ambiente Amazon Virtual Private Cloud (VPC). Se stai eseguendo in un Amazon Virtual Private Cloud, Security Groups (Gruppi di sicurezza) non è disponibile nel pannello di navigazione della console.

Se esegui i nodi di ElastiCache in un Amazon VPC, controlla l'accesso ai cluster con i gruppi di sicurezza di Amazon VPC, che sono diversi dai gruppi di sicurezza di ElastiCache. Per ulteriori informazioni sull'utilizzo di ElastiCache in un Amazon VPC, consulta [Amazon VPC e sicurezza di ElastiCache](#).

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.

Note

Se si prevede di utilizzare [Local Zones](#), assicurati di averlo abilitato. Quando si crea un gruppo di sottorete in quella zona locale, la VPC viene estesa a quella zona locale e la VPC considera la sottorete come qualsiasi sottorete in qualsiasi altra zona di disponibilità. Tutti i gateway e le tabelle di routing pertinenti verranno regolati automaticamente.

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. Fai clic su Gruppi di sicurezza nel riquadro di navigazione.
3. Scegliere o creare un gruppo di sicurezza da utilizzare per le istanze del 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 Redis e i gruppi di replica 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.

Inbound rules (3)									
Filter security group rules									
	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description	
<input type="checkbox"/>	-	sg-...	IPv4	SSH	TCP	22	0.0.0.0/0	-	
<input type="checkbox"/>	-	sg-...	-	Custom TCP	TCP	6379	sg-... / default	-	

Una volta abilitato l'accesso, è ora possibile connettersi al nodo, come illustrato nella sezione successiva.

Per informazioni sull'accesso al cluster ElastiCache da un Amazon VPC diverso, una regione AWS diversa o anche la rete aziendale, consulta:

- [Modelli di accesso a una cache ElastiCache in un Amazon VPC](#)
- [Accesso alle risorse ElastiCache dall'esterno AWS](#)

Fase 4: connessione a un nodo del cluster

Prima di continuare, completa [Fase 3: autorizzazione dell'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 nodo del cluster solo se dispone delle autorizzazioni per eseguire questa operazione.

Individuazione degli endpoint dei nodi

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.

Individuazione degli endpoint di un cluster Redis (modalità cluster disabilitata) (console)

Se un cluster Redis (modalità cluster disabilitata) dispone di un solo nodo, l'endpoint di quest'ultimo viene utilizzato per entrambe le operazioni di lettura e scrittura. Se un cluster dispone di più nodi, ci sono tre tipi di endpoint: l'endpoint primario, l'endpoint di lettura e gli endpoint dei nodi.

L'endpoint primario è un nome DNS che viene sempre risolto nel nodo primario nel cluster. L'endpoint primario è immune a modifiche al cluster, ad esempio la promozione di una replica di lettura al ruolo primario. Per attività di scrittura, è preferibile collegare le applicazioni all'endpoint primario.

Un endpoint di lettura dividerà equamente le connessioni in arrivo sull'endpoint tra tutte le repliche di lettura in un cluster ElastiCache for Redis. Ulteriori fattori come, ad esempio, quando l'applicazione crea le connessioni o il modo in cui l'applicazione riutilizza le connessioni determineranno la distribuzione del traffico. Gli endpoint di lettura mantengono le modifiche ai cluster in tempo reale mano a mano che le repliche vengono aggiunte o rimosse. Puoi posizionare le repliche di lettura multiple del cluster ElastiCache for Redis in diverse zone di disponibilità (AZ) AWS per garantire un'elevata disponibilità degli endpoint di lettura.

Note

Un endpoint di lettura non è un servizio di load balancer. Si tratta di un record DNS che si risolverà in un indirizzo IP di uno dei nodi di replica in modo round robin.




Per attività di lettura, le applicazioni possono inoltre connettersi a qualsiasi nodo nel cluster. A differenza dell'endpoint primario, gli endpoint dei nodi vengono risolti in endpoint specifici. Se apporti una modifica al cluster, ad esempio aggiungendo o eliminando una replica, devi aggiornare gli endpoint dei nodi nell'applicazione.

Per individuare gli endpoint di un cluster Redis (modalità cluster disabilitata)

1. Accedi alla AWS Management Console e apri la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dal pannello di navigazione scegli Cache Redis.

Viene visualizzata la schermata dei cluster con un elenco che include tutte le cache serverless Redis esistenti, i cluster Redis (modalità cluster disabilitata) e Redis (modalità cluster abilitata). Scegliere il cluster creato nella sezione [Creazione di un cluster Redis \(modalità cluster disabilitata\) \(Console\)](#).

3. Per trovare l'endpoint primario e/o l'endpoint di lettura del cluster, scegli il nome del cluster (non il pulsante di opzione).

▼ Cluster details			
Cluster name	Description	Node type	Status
		cache.r6g.large	Available
Engine	Engine version	Global datastore	Global datastore role
Redis	6.0.5	-	-
Update status	Cluster mode	Shards	Number of nodes
Update available	Off	1	3
Data tiering	Multi-AZ	Auto-failover	Encryption in transit
Disabled	Enabled	Enabled	Disabled
Encryption at rest	Parameter group	Outpost ARN	Configuration endpoint
Disabled	default.redis6.x	-	-
Primary endpoint	Reader endpoint	ARN	
 -encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	 -encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com:6379	 -encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	

Endpoint primario e di lettura per un cluster Redis (modalità cluster disabilitata)

Se il cluster annovera un solo nodo, non c'è alcun endpoint primario e si può passare alla fase successiva.

4. Se il cluster Redis (modalità cluster disabilitata) dispone di nodi di replica, puoi trovare gli endpoint del nodo di replica scegliendo il nome del cluster e quindi selezionando la scheda Nodes (Nodi).

Così facendo, viene visualizzata una schermata con l'elenco di tutti i nodi del cluster, primario e di replica, e i relativi endpoint.

<input type="checkbox"/>	Node Name	▲	Status	Current Role	Port	Endpoint
<input type="checkbox"/>	test-no-001		available	primary	6379	test-no-001.usw2.cache.amazonaws.com:6379
<input type="checkbox"/>	test-no-002		available	replica	6379	test-no-002.usw2.cache.amazonaws.com:6379
<input type="checkbox"/>	test-no-003		available	replica	6379	test-no-003.usw2.cache.amazonaws.com:6379

Endpoint di nodo di un cluster Redis (modalità cluster disabilitata)

5. Per copiare un endpoint negli appunti:
 - a. Individua, uno alla volta, gli endpoint da copiare.
 - b. Scegli l'icona di copia direttamente davanti all'endpoint.

L'endpoint è stato correttamente copiato negli appunti. Per informazioni sull'utilizzo dell'endpoint per la connessione a un nodo, vedere [Connessione ai nodi](#).

Un endpoint primario di Redis (modalità cluster disattivata) risulta simile a quello riportato di seguito. Differisce in base all'avvenuta o mancata abilitazione della crittografia dati in transito.

Crittografia dati in transito non abilitata

```
clusterName.xxxxxx.nodeId.regionAndAz.cache.amazonaws.com:port
```

```
redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
```

Crittografia dati in transito abilitata

```
master.clusterName.xxxxxx.regionAndAz.cache.amazonaws.com:port
```

```
master.ncit.ameaqx.use1.cache.amazonaws.com:6379
```

Per trovare gli endpoint, consulta l'argomento pertinente per il motore e il tipo di cluster in esecuzione.

- [Individuazione degli endpoint di connessione](#)
- [Individuazione degli endpoint di un cluster Redis \(modalità cluster abilitata\) \(console\)](#): è necessario l'endpoint di configurazione del cluster.

- [Individuazione degli endpoint \(AWS CLI\)](#)
- [Ricerca degli endpoint \(API ElastiCache\)](#)

Connessione a un cluster Redis o gruppo di replica (Linux)

Ora che disponi dell'endpoint necessario, puoi accedere a un'istanza EC2 e connetterti al cluster o gruppo di replica. Nell'esempio seguente, viene utilizzata l'utilità `redis-cli` per eseguire la connessione a un cluster che esegue Redis. L'ultima versione di `redis-cli` supporta anche SSL/TLS per la connessione di cluster abilitati alla cifra/autenticazione.

Nell'esempio seguente vengono utilizzate istanze Amazon EC2 che eseguono Amazon Linux e Amazon Linux 2. Per dettagli sull'installazione e la compilazione di `redis-cli` con altre distribuzioni Linux, consultare la documentazione relativa al proprio sistema operativo.

Note

Questo processo riguarda il test di una connessione utilizzando l'utilità `redis-cli` solo per uso non pianificato. Per un elenco dei client Redis supportati, consulta la [Documentazione Redis](#). Per esempi di utilizzo dell'AWSSDK con ElastiCache, vedere [Endamenti su ElastiCache e AWS SDK](#).

Connessione a un cluster non crittografato disattivato in modalità cluster

1. Eseguire il comando seguente per connettersi al cluster e sostituire *primary-endpoint* e *port number* con l'endpoint del cluster e il numero di porta. (La porta predefinita per Redis è 6379.)

```
src/redis-cli -h primary-endpoint -p port number
```

Viene generato un prompt dei comandi Redis simile al seguente:

```
primary-endpoint:port number
```

2. Ora puoi eseguire i comandi Redis.

```
set x Hello  
OK
```

```
get x
"Hello"
```

Connessione a un cluster non crittografato attivato in modalità cluster

1. Eseguire il comando seguente per connettersi al cluster e sostituire *configuration-endpoint* e *port number* con l'endpoint del cluster e il numero di porta. (La porta predefinita per Redis è 6379.)

```
src/redis-cli -h configuration-endpoint -c -p port number
```

Note

Nel comando precedente, opzione `-c` abilita la modalità cluster seguente [ASK e -MESSO reindirizzamenti](#).

Viene generato un prompt dei comandi Redis simile al seguente:

```
configuration-endpoint:port number
```

2. Ora puoi eseguire i comandi Redis. Si noti che il reindirizzamento avviene perché è stato abilitato utilizzando l'opzione `-c`. Se il reindirizzamento non è abilitato, il comando restituisce l'errore MESSO. Per ulteriori informazioni sull'errore MESSO, consulta [Specifiche del cluster Redis](#).

```
set x Hi
-> Redirected to slot [16287] located at 172.31.28.122:6379
OK
set y Hello
OK
get y
"Hello"
set z Bye
-> Redirected to slot [8157] located at 172.31.9.201:6379
OK
get z
"Bye"
get x
-> Redirected to slot [16287] located at 172.31.28.122:6379
```



```
"Hi"
```

Connessione a un cluster abilitato per crittografia/autenticazione

Per impostazione predefinita, redis-cli utilizza una connessione TCP non crittografata durante la connessione a Redis. L'opzione `BUILD_TLS=yes` abilita SSL/TLS al momento della compilazione redis-cli, come mostrato nella precedente [Download e configurazione di redis-cli](#) Sezione. L'attivazione di AUTH è facoltativa. Tuttavia, è necessario abilitare la crittografia in transito per abilitare AUTH. Per ulteriori informazioni sulla crittografia ed autenticazione ElastiCache, consulta la [Crittografia in transito ElastiCache \(TLS\)](#).

Note

Puoi utilizzare l'opzione `--tls` con redis-cli per connettersi sia ai cluster crittografati abilitati che disabilitati. Se un cluster dispone di un token AUTH impostato, è possibile utilizzare l'opzione `-a` per fornire una password AUTH.

Negli esempi seguenti, assicurarsi di sostituire *Endpoint del cluster* *Numero di porta* con l'endpoint del cluster e il numero di porta. (La porta predefinita per Redis è 6379.)

Connect ai cluster crittografati disabilitati in modalità cluster

L'esempio seguente si connette a un cluster abilitato per la crittografia e l'autenticazione:

```
src/redis-cli -h cluster-endpoint --tls -a your-password -p port number
```

L'esempio seguente si connette a un cluster per il quale è abilitata solo la crittografia:

```
src/redis-cli -h cluster-endpoint --tls -p port number
```

Connect a cluster crittografati abilitati in modalità cluster

L'esempio seguente si connette a un cluster abilitato per la crittografia e l'autenticazione:

```
src/redis-cli -c -h cluster-endpoint --tls -a your-password -p port number
```

L'esempio seguente si connette a un cluster per il quale è abilitata solo la crittografia:

```
src/redis-cli -c -h cluster-endpoint --tls -p port number
```

Dopo la connessione al cluster, è possibile eseguire i comandi Redis come illustrato negli esempi precedenti per i cluster non crittografati.

Alternativa Redis-CLI

Se il cluster non è abilitato in modalità cluster ed è necessario stabilire una connessione al cluster per un breve test ma senza passare attraverso la compilazione redis-cli, è possibile utilizzare telnet o openssl. Negli esempi seguenti, assicurarsi di sostituire *Endpoint del cluster* e *Numero di porta* con l'endpoint del cluster e il numero di porta. (La porta predefinita per Redis è 6379.)

L'esempio seguente si connette a un cluster abilitato per la crittografia e l'autenticazione:

```
openssl s_client -connect cluster-endpoint:port number
```

Se il cluster dispone di una password impostata, prima connettersi al cluster. Dopo la connessione, autenticare il cluster utilizzando il seguente comando, quindi premere il pulsante Enter chiave. Nell'esempio seguente, sostituire *la password* con la password per il cluster.

```
Auth your-password
```

L'esempio seguente si connette a un cluster disabilitato in modalità cluster che non dispone di crittografia o autenticazione abilitata:

```
telnet cluster-endpoint port number
```

Connessione a un cluster Redis o gruppo di replica (Windows)

Per la connessione al cluster Redis da un'istanza di Windows EC2 utilizzando la CLI di Redis, è necessario scaricare il pacchetto redis-cli e utilizzare redis-cli.exe per la connessione al cluster Redis da un'istanza di Windows EC2.

Nell'esempio seguente, utilizza l'utilità redis-cli per connetterti a un cluster che non è abilitato per la crittografia e che esegue Redis. Per ulteriori informazioni su Redis e i comandi Redis disponibili, consulta la pagina Web dei [comandi Redis](#) sul sito Web di Redis.

Per connetterti a un cluster Redis che non è abilitato per la crittografia utilizzando redis-cli

1. Connettersi all'istanza Amazon EC2 utilizzando l'utilità di connessione preferita. Per istruzioni su come connetterti a un'istanza Amazon EC2, consulta la [Guida alle operazioni di base di Amazon EC2](#).
2. Copiare e incollare il collegamento <https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.zip> in un browser Internet per scaricare il file zip per il client Redis dalla versione disponibile su GitHub <https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504>

Estrarre il file zip nella cartella/percorso desiderato.

Aprire il prompt dei comandi e passare alla directory Redis ed eseguire il comando c :
`\Redis>redis-cli -h Redis_Cluster_Endpoint -p 6379.`

Esempio:

```
c:\Redis>redis-cli -h cmd.xxxxxxx.ng.0001.usw2.cache.amazonaws.com -p 6379
```

3. Eseguire i comandi Redis.

Sei ora connesso al cluster e puoi eseguire comandi Redis del tipo seguente.

```
set a "hello"           // Set key "a" with a string value and no expiration
OK
get a                   // Get value for key "a"
"hello"
get b                   // Get value for key "b" results in miss
(nil)
set b "Good-bye" EX 5  // Set key "b" with a string value and a 5 second expiration
"Good-bye"
get b                   // Get value for key "b"
"Good-bye"

                        // wait >= 5 seconds

get b
(nil)                   // key has expired, nothing returned
quit                    // Exit from redis-cli
```

Passaggio 5: 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 elimini un cluster ElastiCache for Redis, vengono conservati i tuoi snapshot manuali. È inoltre possibile creare uno snapshot finale prima che il cluster venga eliminato. Gli snapshot di cache automatici non vengono conservati. Per ulteriori informazioni, consulta [Snapshot e ripristino](#).

Utilizzo di 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. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di controllo della console ElastiCache, scegliere Redis.

Viene visualizzato l'elenco di tutte le cache che eseguono Redis.

3. Per scegliere il cluster da eliminare, seleziona il nome del cluster nell'elenco di cluster. In questo caso, sostituire con il nome del cluster creato in [Fase 2: creazione di un cluster](#).

Important

Puoi eliminare un solo cluster alla volta dalla console ElastiCache. La selezione di più cluster disabilita l'operazione di eliminazione.

4. In Actions (Azioni), scegliere Delete (Elimina).
5. Nella schermata di conferma dell'eliminazione del cluster, digita il nome del cluster e scegli Backup finale. Quindi seleziona Elimina per eliminare il cluster o 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.

Utilizzo di AWS CLI

Il seguente codice elimina il cluster di cache `my-cluster`. In questo caso, sostituisci `my-cluster` con il nome del cluster creato in [Fase 2: creazione di un cluster](#).

```
aws elasticache delete-cache-cluster --cache-cluster-id my-cluster
```

L'operazione CLI `delete-cache-cluster` elimina un solo cluster di cache. Per eliminare più cluster di cache, chiamare l'operazione `delete-cache-cluster` per ogni cluster da eliminare. Non occorre attendere la fine dell'eliminazione di un cluster di cache prima di eliminarne un altro.

Per Linux, macOS o Unix:

```
aws elasticache delete-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --region us-east-2
```

Per Windows:

```
aws elasticache delete-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --region us-east-2
```

Per ulteriori informazioni, consulta l'argomento AWS CLI per ElastiCache [delete-cache-cluster](#).

Esercitazioni e video relativi a ElastiCache

I tutorial seguenti illustrano le attività di interesse per l'utente di Amazon ElastiCache.

- [Video relativi a ElastiCache](#)
- [Tutorial: Configurazione di una funzione Lambda per l'accesso a Amazon ElastiCache in un Amazon VPC](#)

Video relativi a ElastiCache

Di seguito sono riportati i video che consentono di apprendere concetti di base e avanzati relativi a Amazon ElastiCache. Per informazioni sulla formazione di AWS, vedere [Formazione e Certificazione di AWS](#).

Argomenti

- [Video introduttivi](#)
- [Video avanzati](#)

Video introduttivi

I video seguenti rappresentano un'introduzione ad Amazon ElastiCache.

Argomenti

- [AWS re:Invent 2020: Cosa c'è di nuovo in Amazon ElastiCache](#)
- [AWS re:Invent 2019: Cosa c'è di nuovo in Amazon ElastiCache](#)
- [AWS re:Invent 2017: Cosa c'è di nuovo in Amazon ElastiCache](#)
- [DAT204—Costruire applicazioni scalabili su AWS Servizi NoSQL \(re:Invent 2015\)](#)
- [DAT207—Accelerare le prestazioni delle applicazioni con Amazon ElastiCache \(AWS re:Invent 2013\)](#)

AWS re:Invent 2020: Cosa c'è di nuovo in Amazon ElastiCache

[AWS re:Invent 2020: Cosa c'è di nuovo in Amazon ElastiCache](#)

AWS re:Invent 2019: Cosa c'è di nuovo in Amazon ElastiCache

[AWS re:Invent 2019: Cosa c'è di nuovo in Amazon ElastiCache](#)

AWS re:Invent 2017: Cosa c'è di nuovo in Amazon ElastiCache

[AWS re:Invent 2017: Cosa c'è di nuovo in Amazon ElastiCache](#)

DAT204—Costruire applicazioni scalabili su AWS Servizi NoSQL (re:Invent 2015)

In questa sessione illustreremo i vantaggi dei database NoSQL e forniremo una panoramica dei principali servizi NoSQL offerti da AWS: Amazon DynamoDB e Amazon ElastiCache. Quindi,

ascolteremo gli esponenti di due principali aziende nostre clienti, Expedia e Mapbox, che ci parleranno dei loro casi d'uso, dei problemi legati all'architettura e del modo in cui hanno risolto tali problemi utilizzando i servizi AWS NoSQL, inclusi modelli di progettazione e best practice. Al termine di questa sessione avrai una conoscenza più approfondita di NoSQL e delle sue potenti funzionalità, sarai pronto ad affrontare con sicurezza le sfide del tuo database.

[DAT204—Costruire applicazioni scalabili su Servizi NoSQL AWS\(re:Invent 2015\)](#)

[DAT207—Accelerare le prestazioni delle applicazioni con Amazon ElastiCache \(AWS re:Invent 2013\)](#)

In questo video, scoprirai come utilizzare Amazon ElastiCache per distribuire facilmente un sistema di cache in memoria per velocizzare le prestazioni dell'applicazione. Ti illustreremo le modalità di utilizzo di Amazon ElastiCache per migliorare la latenza dell'applicazione e ridurre il carico sui server del database. Inoltre, ti mostreremo come costruire un livello di caching facile da gestire e dimensionare man mano che l'applicazione cresce. Durante questa sessione, esamineremo diversi scenari e casi d'uso che potrebbero trarre vantaggio dall'attivazione del caching e discuteremo delle caratteristiche di Amazon ElastiCache.

[DAT207 - Accelerare le prestazioni delle applicazioni con Amazon ElastiCache \(re:Invent 2013\)](#)

Video avanzati

I seguenti video trattano argomenti più avanzati relativi ad Amazon ElastiCache.

Argomenti

- [Progettare per il successo con le migliori pratiche di Amazon ElastiCache \(re:Invent 2020\)](#)
- [Supercarica le tue applicazioni in tempo reale con Amazon ElastiCache \(re:Invent 2019\)](#)
- [Migliori pratiche: migrazione di cluster Redis da Amazon EC2 a ElastiCache \(re:Invent 2019\)](#)
- [Dimensionare una piattaforma di Fantasy Sports con Amazon ElastiCache e Amazon Aurora STP11 \(re:Invent 2018\)](#)
- [Redis affidabile e scalabile nel cloud con Amazon ElastiCache \(re:Invent 2018\)](#)
- [Approfondimento di ElastiCache: Modelli di progettazione per archivi di dati in memoria \(re:Invent 2018\)](#)
- [DAT305—Approfondimento Amazon ElastiCache \(re:Invent 2017\)](#)
- [DAT306—Approfondimento Amazon ElastiCache \(re:Invent 2016\)](#)
- [DAT317—How IFTTT uses ElastiCache for Redis to Predict Events \(re:Invent 2016\)](#)

- [DAT407 - Approfondimento Amazon ElastiCache \(Re:Invent 2015\)](#)
- [SDD402—Approfondimento Amazon ElastiCache \(re:Invent 2014\)](#)
- [DAT307—Approfondimento dell'architettura e dei modelli di progettazione di Amazon ElastiCache \(re:Invent 2013\)](#)

Progettare per il successo con le migliori pratiche di Amazon ElastiCache (re:Invent 2020)

Con la crescita esplosiva delle applicazioni in tempo reale di importanza primaria per il business basate su Redis, disponibilità, scalabilità e sicurezza sono diventate le principali considerazioni. Scopri le best practice per la configurazione di Amazon ElastiCache per il successo con dimensionamento online, disponibilità elevata nelle distribuzioni Multi-AZ e configurazioni di sicurezza.

[Progettare per il successo con le migliori pratiche di Amazon ElastiCache \(re:Invent 2020\)](#)

Supercarica le tue applicazioni in tempo reale con Amazon ElastiCache (re:Invent 2019)

Con la rapida crescita dell'adozione del cloud e i nuovi scenari che offre, le applicazioni richiedono una latenza di microsecondi e una velocità effettiva elevata per supportare milioni di richieste al secondo. Gli sviluppatori si sono tradizionalmente affidati a hardware specializzati e soluzioni alternative, come database basati su disco combinati con tecniche di riduzione dei dati, per gestire i dati per applicazioni in tempo reale. Questi approcci possono essere costosi e non scalabili. Scopri come migliorare le prestazioni delle applicazioni in tempo reale utilizzando Amazon ElastiCache completamente gestito e in memoria per prestazioni estreme, scalabilità elevata, disponibilità e sicurezza.

[Supercarica le tue applicazioni in tempo reale con Amazon ElastiCache \(re:Invent 2019\):](#)

Migliori pratiche: migrazione di cluster Redis da Amazon EC2 a ElastiCache (re:Invent 2019)

Gestire i cluster di Redis da soli può essere difficile. È necessario effettuare il provisioning di hardware, software di patch, eseguire il backup dei dati e monitorare costantemente i carichi di lavoro. Con la funzionalità di migrazione online appena rilasciata per Amazon ElastiCache, è ora possibile spostare facilmente i dati da Redis auto-ospitato su Amazon EC2 a Amazon ElastiCache completamente gestito, con la modalità cluster disabilitata. In questa sessione, imparerai a conoscere il nuovo strumento di migrazione online, consultare una demo e, cosa più importante, apprendrai le best practice concrete per una migrazione fluida ad Amazon ElastiCache.

[Migliori pratiche: migrazione di cluster Redis da Amazon EC2 a ElastiCache \(re:Invent 2019\)](#)

[Dimensionare una piattaforma di Fantasy Sports con Amazon ElastiCache e Amazon Aurora STP11 \(re:Invent 2018\)](#)

Dream11 è la startup leader in India nel settore dello sport. Ha una base crescente di oltre 40 milioni di utenti che praticano più sport, tra cui fantasy cricket, calcio e basket, e attualmente serve un milione di utenti simultanei, che producono tre milioni di richieste al minuto sotto un tempo di risposta di 50 millisecondi. In questo discorso, il CTO di Dream11 Amit Sharma spiega come l'azienda utilizza Amazon Aurora e Amazon ElastiCache per gestire il traffico flash, che può triplicare all'interno di una finestra di risposta di 30 secondi. Sharma parla anche di dimensionare le transazioni senza bloccare e condividere i passaggi per gestire il traffico flash, servendo così cinque milioni di utenti attivi giornalmente. Titolo completo: AWS re:Invent 2018: Scalare una piattaforma di Fantasy Sports con Amazon ElastiCache e Amazon Aurora (STP11)

[Dimensionare una piattaforma di Fantasy Sports con Amazon ElastiCache e Amazon Aurora STP11 \(re:Invent 2018\)](#)

[Redis affidabile e scalabile nel cloud con Amazon ElastiCache \(re:Invent 2018\)](#)

Questa sessione tratta le funzionalità e i miglioramenti del nostro servizio compatibile con Redis, Amazon ElastiCache for Redis. Trattiamo funzioni chiave, come Redis 5, scalabilità e miglioramenti delle prestazioni, sicurezza e conformità e molto altro ancora. Discutiamo anche le funzionalità future e i casi di studio dei clienti.

[Redis affidabile e scalabile nel cloud con Amazon ElastiCache \(re:Invent 2018\)](#)

[Approfondimento di ElastiCache: Modelli di progettazione per archivi di dati in memoria \(re:Invent 2018\)](#)

In questa sessione, forniamo uno sguardo dietro le quinte per conoscere la progettazione e l'architettura di Amazon ElastiCache. Vedi i modelli di progettazione comuni con le nostre offerte Redis e Memcached e scopri come i clienti li utilizzano per l'elaborazione dei dati in memoria per ridurre la latenza e migliorare la velocità effettiva delle applicazioni. Rivedremo le best practice, i modelli e gli anti-modelli di progettazione di ElastiCache.

[Approfondimento di ElastiCache: Modelli di progettazione per archivi di dati in memoria \(re:Invent 2018\)](#)

[DAT305—Approfondimento Amazon ElastiCache \(re:Invent 2017\)](#)

Guarda dietro le quinte per scoprire la progettazione e l'architettura di Amazon ElastiCache. Vedi i modelli di progettazione comuni con le nostre offerte Memcached e Redis e scopri come i clienti

li hanno utilizzati per le operazioni in-memory per ridurre la latenza e migliorare il throughput delle applicazioni. Nel corso di questo video, rivedremo le best practice, i modelli e gli anti-modelli di progettazione di ElastiCache.

Il video introduce quanto segue:

- ElastiCache per il resharding online di Redis
- Sicurezza e crittografia in ElastiCache
- ElastiCache for Redis versione 3.2.10

[DAT305—Approfondimento Amazon ElastiCache \(re:Invent 2017\)](#)

DAT306—Approfondimento Amazon ElastiCache (re:Invent 2016)

Guarda dietro le quinte per scoprire la progettazione e l'architettura di Amazon ElastiCache. Vedi i modelli di progettazione comuni con le nostre offerte Memcached e Redis e scopri come i clienti li hanno utilizzati per le operazioni in-memory per ridurre la latenza e migliorare il throughput delle applicazioni. Nel corso di questa sessione, rivedremo le best practice, i modelli e gli anti-modelli di progettazione correlati di ElastiCache.

[DAT306—Approfondimento Amazon ElastiCache \(re:Invent 2016\)](#)

DAT317—How IFTTT uses ElastiCache for Redis to Predict Events (re:Invent 2016)

IFTTT è un servizio gratuito che consente alle persone di sfruttare al meglio i servizi preferiti, dalle semplici attività di automazione alla trasformazione dell'interazione e del controllo delle loro abitazioni. IFTTT utilizza ElastiCache per Redis per archiviare la cronologia di esecuzione delle transazioni e pianificare previsioni nonché indici per registrare i documenti su Amazon S3. Visualizza questa sezione per apprendere come il potere dello scripting di LUA e i tipi di dati di Redis consentivano alle persone di realizzare un progetto che non avrebbero mai potuto realizzare altrimenti.

[DAT317—Come IFTTT usa ElastiCache for Redis per prevedere gli eventi \(re:Invent 2016\)](#)

DAT407 - Approfondimento Amazon ElastiCache (Re:Invent 2015)

Guarda dietro le quinte per scoprire la progettazione e l'architettura di Amazon ElastiCache. Vedi i modelli di progettazione comuni delle nostre offerte Memcached e Redis e scopri come i clienti li hanno utilizzati per le operazioni in-memory e per migliorare la latenza potenziata e il throughput delle

applicazioni. Nel corso di questa sessione, rivedremo le best practice, i modelli e gli anti-modelli di progettazione correlati di Amazon ElastiCache.

[DAT407—Approfondimento Amazon ElastiCache \(re:Invent 2015\)](#)

[SDD402—Approfondimento Amazon ElastiCache \(re:Invent 2014\)](#)

In questo video esamineremo i comuni casi di utilizzo del caching, i motori Memcached e Redis, i modelli che ti consentono di determinare quale modello è migliore per le tue esigenze, l'hashing coerente e molto altro. Tutti elementi che consentono di creare applicazioni veloci e scalabili. Frank Wiebe, principale scienziato in Adobe, illustra come Adobe utilizza Amazon ElastiCache per migliorare l'esperienza cliente e dimensionare il business.

[DAT402—Approfondimento Amazon ElastiCache \(re:Invent 2014\)](#)

[DAT307—Approfondimento dell'architettura e dei modelli di progettazione di Amazon ElastiCache \(re:Invent 2013\)](#)

In questo video esamineremo caching, strategie di caching, dimensionamento e monitoraggio. Confronteremo, inoltre, i motori Memcached e Redis. Nel corso di questa sessione, rivedremo anche le best practice e i modelli di progettazione correlati ad Amazon ElastiCache.

[DAT307 - Approfondimento dell'architettura e dei modelli di progettazione di Amazon ElastiCache \(AWS re:Invent 2013\).](#)

A questo punto come si può procedere?

Ora che hai provato l'esercizio sulle Nozioni di base, puoi esplorare le sezioni seguenti per ottenere ulteriori informazioni su ElastiCache e gli strumenti disponibili:

- [Nozioni di base su AWS](#)
- [Strumenti per Amazon Web Services](#)
- [Interfaccia a riga di comando AWS](#)
- [Informazioni di riferimento API Amazon ElastiCache](#)

Dopo aver completato l'esercizio sulle Nozioni di base, puoi leggere queste sezioni per ottenere ulteriori informazioni sull'amministrazione di ElastiCache:

- [Scelta delle dimensioni dei nodi](#)

Le dimensioni della cache devono essere sufficienti a contenere tutti i dati che desideri memorizzare nella cache. Allo stesso tempo, non vuoi pagare per cache non necessaria. Utilizza questo argomento per scegliere le dimensioni nodo ottimali.

- [Best practice e strategie di caching ElastiCache](#)

Identifica e risolvi i problemi che possono influire sull'efficienza del cluster.

Gestione di nodi

Un nodo è l'elemento base di una distribuzione Amazon ElastiCache. Corrisponde a un blocco di dimensioni fisse di RAM collegata alla rete. Ogni nodo esegue il motore che è stato scelto al momento della creazione e dell'ultima modifica del cluster o del gruppo di replica. Ogni nodo dispone del proprio nome Domain Name Service (DNS) e porta. Sono supportati più tipi di nodi ElastiCache, ciascuno con quantità variabili di memoria associata e potenza di calcolo.

In linea generale, a causa del support per lo sharding, le distribuzioni Redis (modalità cluster abilitata) sono costituite da una serie di nodi più piccoli. Al contrario, le distribuzioni Redis (modalità cluster disabilitata) presentano un numero minore di nodi di maggiori dimensioni. Per una discussione dettagliata sulle dimensioni del nodo da utilizzare, consultare [Scelta delle dimensioni dei nodi](#).

Argomenti

- [Visualizzazione dello stato ElastiCache del nodo](#)
- [Nodi e partizioni Redis](#)
- [Connessione ai nodi](#)
- [Tipi di nodi supportati](#)
- [Riavvio dei nodi \(solo modalità cluster disabilitata\)](#)
- [Sostituzione dei nodi](#)
- [Nodi riservati ElastiCache](#)
- [Migrazione dei nodi della generazione precedente](#)

Alcune importanti operazioni che coinvolgono i nodi sono le seguenti:

- [Aggiunta di nodi a un cluster](#)
- [Rimozione di nodi da un cluster](#)

- [Scalabilità per Redis ElastiCache](#)
- [Individuazione degli endpoint di connessione](#)

Visualizzazione dello stato ElastiCache del nodo

Utilizzando la [ElastiCache console](#), puoi accedere rapidamente allo stato del tuo ElastiCache nodo. Lo stato di un ElastiCache nodo indica lo stato del nodo. Puoi utilizzare le seguenti procedure per visualizzare lo stato del ElastiCache nodo nella ElastiCache console Amazon, il AWS CLI comando o il funzionamento dell'API.

I possibili valori di stato per ElastiCache i nodi sono riportati nella tabella seguente. Questa tabella mostra anche se ti verrà addebitato il costo del ElastiCache nodo.

Type	Fatturato	Descrizione
available	Fatturato	Il ElastiCache nodo è integro e disponibile.
creating	Non fatturata	Il ElastiCache nodo è in fase di creazione. Non è possibile accedere al nodo mentre è in fase di creazione.
deleting	Non fatturata	Il ElastiCache nodo viene eliminato.
modifying	Fatturato	Il ElastiCache nodo viene modificato a causa di una richiesta del cliente di modificare il nodo.
updating	Fatturato	Uno stato di aggiornamento indica che una o più delle seguenti condizioni sono vere per il ElastiCache nodo Amazon:

Type	Fatturato	Descrizione
		<ul style="list-style-type: none">• Il ElastiCache nodo viene patchato come parte dell'aggiornamento del servizio. Per ulteriori informazioni sugli aggiornamenti del servizio, consulta la pagina di aiuto di Amazon ElastiCache Managed Maintenance and Service Updates.• I gruppi di sicurezza VPC si stanno aggiornando per il ElastiCache Cluster.• Il ElastiCache cluster viene scalato verso l'alto o verso il basso.• Le configurazioni di consegna dei log sono in fase di modifica per il Cluster. ElastiCache• Un'operazione di eliminazione per il ElastiCache nodo è in sospeso.• La password ElastiCache per Redis viene aggiornata/ruotata utilizzando. AWS Secrets Manager

Type	Fatturato	Descrizione
rebooting cache cluster nodes	Fatturato	Il ElastiCache nodo viene riavviato a causa di una richiesta del cliente o di un ElastiCache processo Amazon che richiede il riavvio del nodo.
incompatible_parameters	Non fatturata	Amazon non ElastiCache può avviare il nodo perché i parametri specificati nel gruppo di parametri del nodo non sono compatibili con il nodo. Annulla le modifiche apportate ai parametri o rendi tali parametri compatibili con il nodo per ottenere di nuovo l'accesso al nodo. Per ulteriori informazioni sui parametri incompatibili, consulta l'elenco degli eventi per il ElastiCache nodo.

Type	Fatturato	Descrizione
<code>incompatible_network</code>	Non fatturata	<p>Uno stato di rete incompatibile indica che una o più delle seguenti condizioni sono vere per il nodo Amazon: ElastiCache</p> <ul style="list-style-type: none">• Non ci sono indirizzi IP disponibili nella sottorete in cui è stato avviato il ElastiCache nodo.• La sottorete menzionata nel gruppo di ElastiCache sottoreti non esiste più in Amazon Virtual Private Cloud (Amazon VPC).

Type	Fatturato	Descrizione
restore_failed	Non fatturata	<p>Uno stato di ripristino non riuscito indica che per il nodo Amazon ElastiCache si verifica una delle seguenti condizioni:</p> <ul style="list-style-type: none">• Le sostituzioni del nodo non sono riuscite a causa della capacità insufficiente dell'istanza registrata a ripetutamente. Ciò si verifica in genere quando si eseguono nodi di generazione precedente che lo sono. end-of-life Tuttavia, potrebbe verificarsi anche in caso di sostituzione dei nodi della generazione attuale quando AWS non si dispone di una capacità on-demand sufficiente per soddisfare la richiesta nella zona di disponibilità specificata. Per ulteriori informazioni sulla correzione o la rimozione di questi nodi, vedere Migrazione dei nodi della generazione precedente.• Lo snapshot RDB specifico non è stato ripristinato.• L'AWS account per il ElastiCache cluster è stato sospeso.

Type	Fatturato	Descrizione
		<ul style="list-style-type: none">• Il nodo non è riuscito e non può essere ripristinato.
snapshotting	Fatturato	ElastiCache sta creando un'istantanea del nodo ElastiCache for Redis.

Visualizzazione dello stato ElastiCache del nodo con la console

Per visualizzare lo stato di un ElastiCache nodo con la console:

1. Accedi AWS Management Console e apri la ElastiCache console Amazon all'[indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Nel pannello di navigazione, scegli Redis Clusters o Memcached Clusters. Viene visualizzata la pagina Cache con l'elenco dei nodi. ElastiCache Per ogni nodo viene visualizzato il valore dello stato.
3. È quindi possibile accedere alla scheda Service Updates relativa alla cache per visualizzare l'elenco degli aggiornamenti del servizio applicabili alla cache.

Visualizzazione dello stato del ElastiCache nodo con AWS CLI

Per visualizzare il ElastiCache nodo e le relative informazioni sullo stato utilizzando il AWS CLI, usa il `describe-cache-cluster` comando. Ad esempio, il AWS CLI comando seguente visualizza ogni ElastiCache nodo.

```
aws elasticache describe-cache-clusters
```

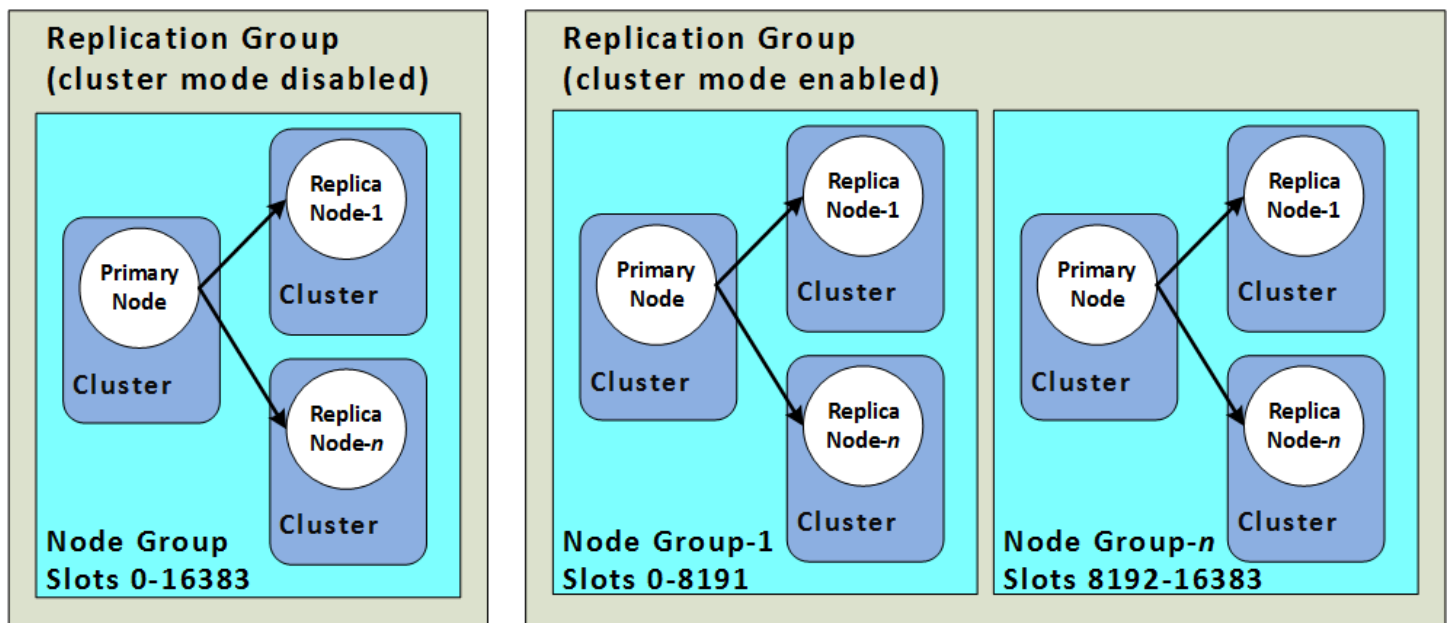
Visualizzazione dello stato del ElastiCache nodo tramite l'API

Per visualizzare lo stato del ElastiCache nodo utilizzando l' ElastiCache API Amazon, chiamalo `DescribeCacheClusteroperation` con il `ShowCacheNodeInfo` flag per recuperare informazioni sui singoli nodi di cache.

Nodi e partizioni Redis

Una partizione (nelle API e nella CLI, un gruppo di nodi) è una disposizione gerarchica di nodi, ciascuno integrato 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. Redis versione 3.2 e successive supportano più partizioni all'interno di un cluster (nelle API e nella CLI, un gruppo di replica). Questo support consente il partizionamento dei dati in un cluster Redis (modalità cluster abilitata).

Il seguente diagramma mostra le differenze tra un cluster Redis (modalità cluster disabilitata) e un cluster Redis (modalità cluster abilitata).



I cluster Redis (modalità cluster abilitata) supportano la replica tramite partizioni. L'operazione API, [DescribeReplicationGroups](#) (CLI: [describe-replication-groups](#)) elenca i gruppi di nodi con i nodi membro, il ruolo del nodo all'interno del gruppo di nodi nonché altre informazioni.

Quando si crea un cluster Redis, si specifica se si desidera creare un cluster con clustering abilitato. I cluster disabilitati in modalità cluster Redis non contengono mai più di una partizione che può essere dimensionato orizzontalmente aggiungendo (fino a un totale di cinque) o eliminando nodi di replica di lettura. Per ulteriori informazioni, consultare [Alta disponibilità utilizzando gruppi di replica](#), [Aggiunta di una replica di lettura per i gruppi di replica Redis \(modalità cluster disabilitata\)](#) o [Eliminazione di una replica di lettura, per i gruppi di replica Redis \(modalità cluster disabilitata\)](#). I cluster Redis (modalità cluster disabilitata) possono anche essere dimensionati verticalmente modificando i tipi di nodi. Per ulteriori informazioni, consulta [Ridimensionamento dei cluster Redis \(modalità cluster disabilitata\) con nodi di replica](#).

Il limite del nodo o della partizione può essere aumentato fino a un massimo di 500 per cluster se la versione del motore Redis è 5.0.6 o superiore. Ad esempio, è possibile scegliere di configurare un cluster a 500 nodi che varia tra 83 partizioni (un primario e 5 repliche per partizione) e 500 partizioni (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. Per ulteriori informazioni, consulta [Creazione di un gruppo di sottoreti](#).

Per le versioni inferiori alla 5.0.6, il limite è 250 per cluster.

Per richiedere un aumento dei limiti, consulta [AWS Limiti dei servizi](#) e seleziona il tipo di limite Nodi per cluster per tipo di istanza.

Un cluster Redis (modalità cluster abilitata) può essere modificato (adattato in eccesso o in difetto) dopo essere stato creato. Per ulteriori informazioni, consulta [Scalabilità per Redis ElastiCache](#) 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. Tale approccio funziona solo se il gruppo del cluster presenta lo stesso numero di partizioni del cluster precedente. Ciò può essere utile se occorre modificare il tipo di nodo o la versione dell'engine. Per ulteriori informazioni, consulta [Esecuzione di backup manuali](#) e [Ripristino da un backup in una nuova cache](#).

Connessione ai nodi

Prima di provare a connettersi ai nodi nel cluster Redis, sono necessari gli endpoint per i nodi. Per individuare gli endpoint, consultare la seguente documentazione:

- [Individuazione degli endpoint di un cluster Redis \(modalità cluster disabilitata\) \(console\)](#)
- [Individuazione degli endpoint di un cluster Redis \(modalità cluster abilitata\) \(console\)](#)
- [Individuazione degli endpoint \(AWS CLI\)](#)
- [Ricerca degli endpoint \(API ElastiCache\)](#)

Nell'esempio seguente, viene utilizzata l'utilità redis-cli per eseguire la connessione a un cluster che esegue Redis.

Note

Per ulteriori informazioni su Redis e i comandi Redis disponibili, consulta la pagina Web all'indirizzo <http://redis.io/commands>.

Per connettersi a un cluster Redis mediante redis-cli

1. Connettersi all'istanza Amazon EC2 utilizzando l'utilità di connessione preferita.

Note

Per istruzioni su come connetterti a un'istanza Amazon EC2, consulta la [Guida alle operazioni di Amazon EC2](#).

2. Per compilare redis-cli, scaricare e installare la GNU Compiler Collection (gcc). Dal prompt dei comandi dell'istanza EC2, digitare il seguente comando e digitare y al prompt dei comandi.

```
sudo yum install gcc
```

Viene visualizzato un output simile al seguente.

```
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
```

```
--> Running transaction check

...(output omitted)...

Total download size: 27 M
Installed size: 53 M
Is this ok [y/N]: y
Downloading Packages:
(1/11): binutils-2.22.52.0.1-10.36.amzn1.x86_64.rpm      | 5.2 MB    00:00
(2/11): cpp46-4.6.3-2.67.amzn1.x86_64.rpm             | 4.8 MB    00:00
(3/11): gcc-4.6.3-3.10.amzn1.noarch.rpm               | 2.8 kB    00:00

...(output omitted)...

Complete!
```

3. Scaricare e compilare l'utilità redis-cli. Questa utilità è inclusa nella distribuzione del software Redis. Dal prompt dei comandi dell'istanza EC2, digitare i seguenti comandi.

Note

Per sistemi Ubuntu, prima di eseguire make, esegui `make distclean`.

```
wget http://download.redis.io/redis-stable.tar.gz
tar xvzf redis-stable.tar.gz
cd redis-stable
make distclean      # ubuntu systems only
make
```

4. Dal prompt dei comandi dell'istanza EC2, digitare i seguenti comandi.

```
src/redis-cli -c -h mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com -p 6379
```

Viene visualizzato un prompt dei comandi Redis simile al seguente.

```
redis mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com 6379>
```

5. Testare la connessione eseguendo i comandi Redis.

Una volta connessi al cluster è possibile eseguire i comandi Redis. Di seguito sono riportati alcuni comandi di esempio con le risposte Redis.

```
set a "hello"           // Set key "a" with a string value and no expiration
OK
get a                   // Get value for key "a"
"hello"
get b                   // Get value for key "b" results in miss
(nil)
set b "Good-bye" EX 5  // Set key "b" with a string value and a 5 second expiration
get b
"Good-bye"

                        // wait 5 seconds

get b
(nil)                   // key has expired, nothing returned
quit                   // Exit from redis-cli
```

Per la connessione ai nodi o ai cluster con crittografia SSL (Secure Sockets Layer) abilitata in transito, consulta [Crittografia in transito ElastiCache \(TLS\)](#).

Tipi di nodi supportati

ElastiCache supporta i seguenti tipi di nodi. In generale, le generazioni attuali offrono più memoria e più potenza di elaborazione a un costo inferiore rispetto ai prodotti omologhi della generazione precedente.

Per ulteriori informazioni sui dettagli sulle prestazioni per ogni tipo di nodo, consulta [Tipi di istanza di Amazon EC2](#).

Per informazioni sulle dimensioni del nodo da utilizzare, consultare [Scelta delle dimensioni dei nodi](#).

Generazione attuale

Per ulteriori informazioni sulla generazione precedente, consultare [Nodi di generazione precedente](#).

Note

I tipi di istanza con prestazioni delle reti espandibili utilizzano un meccanismo di credito I/O di rete per superare la larghezza di banda di base sulla base del massimo sforzo.

Generale

Tipo di istanza	Versione minima supportata di Redis	I/O avanzato (Redis 5.0.6+)	Offload TLS (Redis 6.2.5+)	Multiple I/O avanzate (Redis 7.0.4+)	Larghezza di banda di base (Gb/s)	Larghezza di banda di burst (Gb/s)
cache.m7g.large	6.2	N	N	N	0,937	12,5
cache.m7g.xlarge	6.2	Y	Y	Y	1,876	12,5
cache.m7g.2xlarge	6.2	Y	Y	Y	3,75	15

Tipo di istanza	Versione minima supportata di Redis	I/O avanzato (Redis 5.0.6+)	Offload TLS (Redis 6.2.5+)	Multiple I/O avanzato (Redis 7.0.4+)	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.m7g.4xlarge	6.2	Y	Y	Y	7,5	15
cache.m7g.8xlarge	6.2	Y	Y	Y	15	N/D
cache.m7g.12xlarge	6.2	Y	Y	Y	22,5	N/D
cache.m7g.16xlarge	6.2	Y	Y	Y	30	N/D
cache.m6g.large	5.0.6	N	N	N	0.75	10,0
cache.m6g.xlarge	5.0.6	Y	Y	Y	1,25	10,0
cache.m6g.2xlarge	5.0.6	Y	Y	Y	2.5	10,0
cache.m6g.4xlarge	5.0.6	Y	Y	Y	5.0	10,0
cache.m6g.8xlarge	5.0.6	Y	Y	Y	12	N/D
cache.m6g.12xlarge	5.0.6	Y	Y	Y	20	N/D
cache.m6g.16xlarge	5.0.6	Y	Y	Y	25	N/D

Tipo di istanza	Versione minima supportata di Redis	I/O avanzato (Redis 5.0.6+)	Offload TLS (Redis 6.2.5+)	Multiple I/O avanzato (Redis 7.0.4+)	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.m5.large	3.2.4	N	N	N	0,75	10,0
cache.m5.xlarge	3.2.4	Y	N	N	1,25	10,0
cache.m5.2xlarge	3.2.4	Y	Y	Y	2,5	10,0
cache.m5.4xlarge	3.2.4	Y	Y	Y	5,0	10,0
cache.m5.12xlarge	3.2.4	Y	Y	Y	12	N/D
cache.m5.24xlarge	3.2.4	Y	Y	Y	25	N/D
cache.m4.large	3.2.4	N	N	N	0,45	1,2
cache.m4.xlarge	3.2.4	Y	N	N	0,75	2,8
cache.m4.2xlarge	3.2.4	Y	Y	Y	1,0	10,0
cache.m4.4xlarge	3.2.4	Y	Y	Y	2,0	10,0
cache.m4.10xlarge	3.2.4	Y	Y	Y	5,0	10,0
cache.t4g.micro	3.2.4	N	N	N	0,064	5,0
cache.t4g.small	5.0.6	N	N	N	0,128	5,0

Tipo di istanza	Versione minima supportata di Redis	I/O avanzato (Redis 5.0.6+)	Offload TLS (Redis 6.2.5+)	Multiple I/O avanzato (Redis 7.0.4+)	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.t4g.medium	5.0.6	N	N	N	0,256	5.0
cache.t3.micro	3.2.4	N	N	N	0,064	5.0
cache.t3.small	3.2.4	N	N	N	0,128	5.0
cache.t3.medium	3.2.4	N	N	N	0,256	5.0
cache.t2.micro	3.2.4	N	N	N	0,064	1,024
cache.t2.small	3.2.4	N	N	N	0,128	1,024
cache.t2.medium	3.2.4	N	N	N	0,256	1,024

Memoria ottimizzata

Tipo di istanza	Versione minima supportata di Redis	I/O avanzato (Redis 5.0.6+)	Offload TLS (Redis 6.2.5+)	Multiple I/O avanzato (Redis 7.0.4+)	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.r7g.large	6.2	N	N	N	0,937	12,5
cache.r7g.xlarge	6.2	Y	Y	Y	1,876	12,5

Tipo di istanza	Versione minima supportata di Redis	I/O avanzato (Redis 5.0.6+)	Offload TLS (Redis 6.2.5+)	Multiple I/O avanzato (Redis 7.0.4+)	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.r7g. .2xlarge	6.2	Y	Y	Y	3,75	15
cache.r7g. .4xlarge	6.2	Y	Y	Y	7,5	15
cache.r7g. .8xlarge	6.2	Y	Y	Y	15	N/D
cache.r7g. .12xlarge	6.2	Y	Y	Y	22,5	N/D
cache.r7g. .16xlarge	6.2	Y	Y	Y	30	N/D
cache.r6g.large	5.0.6	N	N	N	0.75	10,0
cache.r6g.xlarge	5.0.6	Y	Y	Y	1,25	10,0
cache.r6g. .2xlarge	5.0.6	Y	Y	Y	2.5	10,0
cache.r6g. .4xlarge	5.0.6	Y	Y	Y	5.0	10,0
cache.r6g. .8xlarge	5.0.6	Y	Y	Y	12	N/D
cache.r6g. .12xlarge	5.0.6	Y	Y	Y	20	N/D

Tipo di istanza	Versione minima supportata di Redis	I/O avanzato (Redis 5.0.6+)	Offload TLS (Redis 6.2.5+)	Multiple I/O avanzate (Redis 7.0.4+)	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.r6g.16xlarge	5.0.6	Y	Y	Y	25	N/D
cache.r5.large	3.2.4	N	N	N	0.75	10,0
cache.r5.xlarge	3.2.4	Y	N	N	1,25	10,0
cache.r5.2xlarge	3.2.4	Y	Y	Y	2.5	10,0
cache.r5.4xlarge	3.2.4	Y	Y	Y	5.0	10,0
cache.r5.12xlarge	3.2.4	Y	Y	Y	12	N/D
cache.r5.24xlarge	3.2.4	Y	Y	Y	25	N/D
cache.r4.large	3.2.4	N	N	N	0.75	10,0
cache.r4.xlarge	3.2.4	Y	N	N	1,25	10,0
cache.r4.2xlarge	3.2.4	Y	Y	Y	2.5	10,0
cache.r4.4xlarge	3.2.4	Y	Y	Y	5.0	10,0
cache.r4.8xlarge	3.2.4	Y	Y	Y	12	N/D
cache.r4.16xlarge	3.2.4	Y	Y	Y	25	N/D

Memoria ottimizzata con il tiering dei dati

Tipo di istanza	Versione minima supportata di Redis	I/O avanzato (Redis 5.0.6+)	Offload TLS (Redis 6.2.5+)	Multiple I/O avanzate (Redis 7.0.4+)	Larghezza di banda base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.r6gd.xlarge	6.2.0	Y	N	N	1,25	10
cache.r6gd.2xlarge	6.2.0	Y	Y	Y	2.5	10
cache.r6gd.4xlarge	6.2.0	Y	Y	Y	5.0	10
cache.r6gd.8xlarge	6.2.0	Y	Y	Y	12	N/D
cache.r6gd.12xlarge	6.2.0	Y	Y	Y	20	N/D
cache.r6gd.16xlarge	6.2.0	Y	Y	Y	25	N/D

Ottimizzato per la rete

Tipo di istanza	Versione minima supportata di Redis	I/O avanzato (Redis 5.0.6+)	Offload TLS (Redis 6.2.5+)	Multiplexing I/O avanzato (Redis 7.0.4+)	Larghezza di banda di base (Gb/s)	Larghezza di banda di burst (Gb/s)
cache.c7gn.large	6.2	N	N	N	6,25	30
cache.c7gn.xlarge	6.2	Y	Y	Y	12,5	40
cache.c7gn.2xlarge	6.2	Y	Y	Y	25	50
cache.c7gn.4xlarge	6.2	Y	Y	Y	50	N/D
cache.c7gn.8xlarge	6.2	Y	Y	Y	100	N/D
cache.c7gn.12xlarge	6.2	Y	Y	Y	150	N/D
cache.c7gn.16xlarge	6.2	Y	Y	Y	200	N/D

Tipi di nodi supportati in base alla regione AWS

I tipi di nodi supportati possono variare da una regione AWS all'altra. Per ulteriori dettagli, consultare [Prezzi di Amazon ElastiCache](#).

Istanze a prestazioni espandibili

Puoi ora avviare i nodi di cache T4g, T3-Standard e T2-Standard espandibili e di scopo generico su Amazon ElastiCache. Questi nodi offrono un livello base di prestazioni della CPU con la possibilità di

incrementarne l'utilizzo quando necessario fino all'esaurimento dei crediti accumulati. Un credito CPU fornisce le prestazioni di un core CPU completo per un minuto.

I nodi T4g, T3 e T2 di Amazon ElastiCache sono configurati come standard e adatti a carichi di lavoro con un utilizzo medio della CPU che è costantemente al di sotto delle prestazioni di base dell'istanza. Per superare la baseline, il nodo spende i crediti accumulati nel suo saldo del credito CPU. Se il nodo in esecuzione genera un numero basso di crediti, le prestazioni vengono gradualmente ridotte al livello base. Questa riduzione graduale assicura che il nodo non subirà un drastico calo delle prestazioni se il credito della CPU accumulato viene esaurito. Per ulteriori informazioni, consulta [Crediti CPU e prestazioni di base per istanze a prestazioni espandibili](#) nella Guida per l'utente Amazon EC2.

La tabella seguente elenca i tipi di nodi dalle prestazioni espandibili e la velocità oraria con cui vengono ottenuti i crediti CPU. Mostra anche il numero massimo di crediti CPU che un nodo può ottenere e il numero di vCPU per nodo. Inoltre, indica il livello di prestazioni di base come percentuale di prestazioni full core (utilizzando una vCPU singola).

Crediti CPU guadagnati all'ora	Quantità massima di crediti guadagnati che può essere accumulati a*	vCPU	Prestazioni di base per vCPU	Memoria (GiB)	Prestazioni di rete
12	288	2	10%	0,5	Fino a 5 Gigabit
24	576	2	20%	1,37	Fino a 5 Gigabit
24	576	2	20%	3,09	Fino a 5 Gigabit
12	288	2	10%	0,5	Fino a 5 Gigabit

Crediti CPU guadagnati all'ora	Quantità massima di crediti guadagnati che può essere accumulata*	vCPU	Prestazioni di base per vCPU	Memoria (GiB)	Prestazioni di rete
24	576	2	20%	1,37	Fino a 5 Gigabit
24	576	2	20%	3,09	Fino a 5 Gigabit
6	144	1	10%	0,5	Basse a moderate
12	288	1	20%	1,55	Basse a moderate
24	576	2	20%	3,22	Basse a moderata

* Il numero di crediti che possono essere accumulati è equivalente al numero di crediti che possono essere guadagnati in un periodo di 24 ore.

** Le prestazioni di base nella tabella si riferiscono alla singola vCPU. Alcune dimensioni di nodo che hanno più di una vCPU. Per queste, calcolare l'utilizzo della CPU di base per il nodo, moltiplicando la percentuale di vCPU per il numero di vCPU.

Le seguenti metriche di credito CPU sono disponibili per le istanze a prestazioni espandibili T3 e T4:

Note

Queste metriche non sono disponibili per le istanze di prestazioni Burstable T2.

- CPUcreditUsage
- CPUcreditBalance

Per ulteriori informazioni su tali parametri, consulta [Parametri di credito della CPU](#).

Inoltre, ricordare i seguenti dettagli:

- Tutti i tipi di nodo della generazione corrente vengono creati in un cloud privato virtuale (VPC) basato per impostazione predefinita su Amazon VPC.
- I file non modificabile (AOF) Redis non sono supportati per le istanze T2. Le variabili di configurazione Redis `appendonly` e `appendfsync` non sono supportate su Redis versione 2.8.22 e successive.

Informazioni correlate

- [Dettagli e caratteristiche del prodotto Amazon ElastiCache](#)
- [Parametri specifici di Redis](#)
- [Crittografia dei dati in transito \(TLS\)](#)

Riavvio dei nodi (solo modalità cluster disabilitata)

Affinché vengano applicate, alcune modifiche richiedono il riavvio dei cluster di nodi. Ad esempio, per alcuni parametri la modifica del valore in un gruppo di parametri si applica solo dopo un riavvio.

Per cluster Redis (modalità cluster disabilitata), questi parametri sono:

- `activerehashing`
- `database`

Si può riavviare un nodo utilizzando solo la console ElastiCache. È possibile riavviare un solo nodo alla volta. Per riavviare più nodi è necessario ripetere il processo per ogni nodo.

Modificazioni dei parametri Redis (modalità cluster abilitata)

Se apporti modifiche ai seguenti parametri in un cluster Redis (modalità cluster abilitata), segui i passaggi seguenti.

- activerehashing
 - database
1. Creare un backup manuale del cluster. Per informazioni, consultare [Esecuzione di backup manuali](#).
 2. Eliminare il cluster Redis (modalità cluster abilitata). Per informazioni, consultare [Eliminazione di un cluster](#).
 3. Ripristinare il cluster utilizzando il gruppo di parametri modificato e il backup per inizializzare il nuovo cluster. Per informazioni, consultare [Ripristino da un backup in una nuova cache](#).

Le modifiche ad altri parametri non richiedono questo.

Utilizzo di AWS Management Console

È possibile riavviare un nodo utilizzando la console ElastiCache.

Per riavviare un nodo (console)

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dall'elenco nell'angolo in alto a destra, scegliere la regione AWS applicabile.
3. Nel pannello di navigazione a sinistra, scegliere Redis.

Viene visualizzato un elenco dei cluster che eseguono una versione di Redis.

4. Scegliere il cluster in Nome del cluster.
5. In Nome nodo scegliere il pulsante di opzione accanto al nodo che si desidera riavviare.
6. Scegliere Instance actions (Operazioni istanza), quindi Reboot node (Riavvia nodo).

Per riavviare più cluster, ripetere le fasi da 2 a 5 per ciascun nodo da riavviare. Non occorre attendere la fine del riavvio di un nodo prima di riavviane un altro.

Sostituzione dei nodi

Amazon ElastiCache for Redis aggiorna frequentemente il parco istanze con patch e aggiornamenti che vengono applicati a istanze. Tuttavia, di tanto in tanto siamo costretti a lanciare nuovi nodi ElastiCache for Redis per applicare aggiornamenti obbligatori del sistema operativo dell'host. 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 cache sostitutivi generati automaticamente da Amazon ElastiCache potrebbero avere indirizzi IP diversi. L'utente è responsabile della verifica della configurazione dell'applicazione per assicurarsi che i nodi della cache siano associati agli indirizzi IP appropriati.

Nell'elenco seguente vengono identificate le operazioni che puoi eseguire quando ElastiCache pianifica la sostituzione di uno dei nodi Redis. Per velocizzare l'individuazione delle informazioni richieste per questa situazione, scegliere dal seguente menu.

- [Do nothing](#)— Lascia che Amazon ElastiCache sostituisca il nodo come pianificato.
- [Change your maintenance window](#)— Cambia la finestra di manutenzione in un momento migliore.
- Configurazioni Redis (modalità cluster abilitata)
 - [Replace the only node in any Redis cluster](#)— Procedura per sostituire un nodo in un cluster Redis utilizzando il backup e il ripristino.
 - [Replace a replica node in any Redis cluster](#)— Procedura per sostituire una replica di lettura in qualsiasi cluster Redis aumentando e diminuendo il numero di repliche senza tempi di inattività del cluster.
 - [Replace any node in a Redis \(cluster mode enabled\) shard](#)— Una procedura dinamica senza tempi di inattività del cluster per sostituire un nodo in un cluster Redis (modalità cluster abilitata) mediante scalabilità orizzontale e scalabilità in.

- Configurazioni Redis (modalità cluster disabilitata)
 - [Replace the only node in any Redis cluster](#)— Procedura per sostituire qualsiasi nodo in un cluster Redis utilizzando il backup e il ripristino.
 - [Replace a replica node in any Redis cluster](#)— Procedura per sostituire una replica di lettura in qualsiasi cluster Redis aumentando e diminuendo il numero di repliche senza tempi di inattività del cluster.
 - [Replace a node in a Redis \(cluster mode disabled\) cluster](#)— Procedura per sostituire un nodo in un cluster Redis (modalità cluster disabilitata) utilizzando la replica.
 - [Replace a Redis \(cluster mode disabled\) read-replica](#)— Procedura per sostituire manualmente una replica di lettura in un gruppo di replica Redis (modalità cluster disabilitata).
 - [Replace a Redis \(cluster mode disabled\) primary node](#)— Procedura per sostituire manualmente il nodo primario in un gruppo di replica Redis (modalità cluster disabilitata).

Opzioni di sostituzione nodo Redis

- Non fare nulla : Se non fai nulla, ElastiCache sostituirà il nodo come pianificato.

Per configurazioni non cluster con failover automatico abilitato, i cluster su Redis 5.0.6 completano la sostituzione mentre il cluster continua a rimanere online e servire le richieste di scrittura in entrata. Per cluster abilitati per il failover automatico su Redis 4.0.10 o versioni precedenti, si potrebbe notare una breve interruzione della scrittura, fino ad alcuni secondi, associata ad aggiornamenti DNS.

Se il nodo è membro di un cluster con failover automatico, ElastiCache fornisce una migliore disponibilità durante le patch, gli aggiornamenti e altre sostituzioni dei nodi relative alla manutenzione.

Per configurazioni ElastiCache for Redis Cluster impostate per utilizzare client ElastiCache for Redis Cluster, la sostituzione viene completata mentre il cluster server richieste di scrittura.

Per configurazioni non cluster con failover automatico abilitato, i cluster su Redis 5.0.6 completano la sostituzione mentre il cluster continua a rimanere online e servire le richieste di scrittura in

entrata. Per cluster abilitati per il failover automatico su Redis 4.0.10 o versioni precedenti, si potrebbe notare una breve interruzione della scrittura, fino ad alcuni secondi, associata ad aggiornamenti DNS.

Se il nodo è autonomo, Amazon ElastiCache avvia innanzitutto un nodo sostitutivo e quindi esegue la sincronizzazione dal nodo esistente. Durante tale periodo il nodo esistente non sarà disponibile per richieste di servizio. Al termine della sincronizzazione, il nodo esistente viene terminato e il nuovo nodo prende il suo posto. ElastiCache fa il massimo sforzo per conservare i dati durante questa operazione.

- Modifica la finestra di manutenzione - Per gli eventi di manutenzione programmata, ricevi un'e-mail o un evento di notifica da ElastiCache. 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 gli argomenti seguenti:
 - [Modifica di un cluster ElastiCache](#)
 - [Modifica di un gruppo di replica](#)

Note

La possibilità di modificare la finestra di sostituzione spostando la finestra di manutenzione è disponibile solo quando la notifica ElastiCache 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.

- Sposti la finestra di manutenzione alle ore 16:00 di mercoledì, anticipandola rispetto alla data e ora corrente. Il nodo viene sostituito il prossimo mercoledì 15 novembre alle 16:00.

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

- Sostituire l'unico nodo in qualsiasi cluster Redis – Se il cluster non contiene repliche di lettura, è possibile utilizzare la seguente procedura per sostituire il nodo.

Per sostituire l'unico nodo utilizzando backup e ripristino

1. Creare una snapshot del cluster del nodo. Per istruzioni, consultare [Esecuzione di backup manuali](#).
 2. Creare un nuovo cluster inizializzandolo dalla snapshot. Per istruzioni, consultare [Ripristino da un backup in una nuova cache](#).
 3. Eliminare il cluster con il nodo pianificato per la sostituzione. Per istruzioni, consultare [Eliminazione di un cluster](#).
 4. Nell'applicazione, sostituire l'endpoint del vecchio nodo con l'endpoint del nuovo nodo.
- Sostituisci un nodi di replica in qualsiasi cluster Redis Per sostituire un cluster di replica, incrementare il numero di repliche. Per farlo, aggiungere una replica quindi ridurre il numero delle repliche rimuovendo la replica che si desidera sostituire. Questo processo è dinamico e non provoca tempi di inattività del cluster.

Note

Se lo shard o il gruppo di replica dispongono già di repliche, è necessario invertire le fasi 1 e 2.

Per sostituire una replica in qualsiasi cluster Redis

1. Incrementare il conteggio delle repliche aggiungendo una replica allo shard o gruppo di replica. Per ulteriori informazioni, consulta [Incremento del numero di repliche in uno shard](#).
2. Eliminare la replica da sostituire. Per ulteriori informazioni, consulta [Riduzione del numero di repliche in uno shard](#).

3. Aggiornare gli endpoint nell'applicazione.
- Sostituire qualsiasi nodo in una partizione Redis (modalità cluster abilitata) — Per sostituire il nodo in un cluster senza tempi di inattività, utilizza il reshard online. Prima di tutto aggiungere uno shard incrementando le dimensioni, quindi eliminare lo shard con il nodo da sostituire riducendo le dimensioni.

Per sostituire qualsiasi nodo in un cluster Redis (modalità cluster abilitata)

1. Dimensionamento orizzontale: aggiungere un ulteriore shard con la stessa configurazione dello shard esistente con il nodo da sostituire. Per ulteriori informazioni, consulta [Aggiunta delle partizioni con il resharding online](#).
 2. Dimensionamento verticale: eliminare lo shard con il nodo da sostituire. Per ulteriori informazioni, consulta [Rimozione delle partizioni con il resharding online](#).
 3. Aggiornare gli endpoint nell'applicazione.
- Sostituire un nodo in un cluster Redis (modalità cluster disabilitata)— Se il cluster è un cluster Redis (modalità cluster disabilitata) senza repliche di lettura, per sostituire il nodo è possibile utilizzare la seguente procedura.

Per sostituire il nodo utilizzando la replica (solo modalità cluster disabilitata)

1. Aggiungere replica al cluster con il nodo pianificato per la sostituzione come primario. Non abilitare Multi-AZ su questo cluster. Per istruzioni, consultare [Per aggiungere la funzione di replica a un cluster Redis privo di partizioni](#).
2. Aggiungere una replica di lettura al cluster. Per istruzioni, consultare [Per aggiungere nodi a un cluster \(console\)](#).
3. Promuovere la nuova replica di lettura creata al nodo primario. Per istruzioni, consultare [Promozione di una replica di lettura sul primario, per gruppi di replica Redis \(modalità cluster disabilitata\)](#).
4. Eliminare il nodo pianificato per la sostituzione. Per istruzioni, consultare [Rimozione di nodi da un cluster](#).
5. Nell'applicazione, sostituire l'endpoint del vecchio nodo con l'endpoint del nuovo nodo.

- Sostituire una replica di lettura Redis (modalità cluster disabilitata)— Se il nodo è una replica di lettura, sostituire il nodo.

Se il cluster dispone di un solo nodo di replica e Multi-AZ è abilitata, è necessario disabilitare Multi-AZ prima di poter eliminare la replica. Per istruzioni, consultare [Modifica di un gruppo di replica](#).

Per sostituire una replica di lettura Redis (modalità cluster disabilitata)

1. Eliminare la replica che è stata pianificata per la sostituzione. Per le istruzioni, consultare quanto segue:
 - [Riduzione del numero di repliche in uno shard](#)
 - [Rimozione di nodi da un cluster](#)
 2. Aggiungere una nuova replica a quella che è pianificata per la sostituzione. Se si utilizza lo stesso nome della replica appena eliminata, è possibile andare al punto 3. Per le istruzioni, consultare quanto segue:
 - [Incremento del numero di repliche in uno shard](#)
 - [Aggiunta di una replica di lettura per i gruppi di replica Redis \(modalità cluster disabilitata\)](#)
 3. Nell'applicazione, sostituire l'endpoint della vecchia replica con l'endpoint della nuova replica.
 4. Se Multi-AZ è stato inizialmente disabilitato, riabilitarlo ora. Per istruzioni, consultare [Abilitazione della funzione Multi-AZ](#).
-
- Sostituisci un nodo primario (modalità cluster disabilitata) Redis: se il nodo è quello primario, promuovi prima una replica di lettura al ruolo di nodo primario. Eliminare quindi la replica utilizzata precedentemente come nodo primario.

Se il cluster dispone di un solo nodo di replica e Multi-AZ è abilitata, è necessario disabilitare Multi-AZ prima di poter eliminare la replica nel passaggio 2. Per istruzioni, consultare [Modifica di un gruppo di replica](#).

Per sostituire un nodo primario Redis (modalità cluster disabilitata)

1. Promuovere una replica di lettura al nodo primario. Per istruzioni, consultare [Promozione di una replica di lettura sul primario, per gruppi di replica Redis \(modalità cluster disabilitata\)](#).

2. Eliminare il nodo pianificato per la sostituzione (il vecchio nodo primario). Per istruzioni, consultare [Rimozione di nodi da un cluster](#).
3. Aggiungere una nuova replica a quella pianificata per la sostituzione. Se si utilizza lo stesso nome del nodo appena eliminato, è possibile ignorare la modifica degli endpoint nell'applicazione.

Per istruzioni, consultare [Aggiunta di una replica di lettura per i gruppi di replica Redis \(modalità cluster disabilitata\)](#).

4. Nell'applicazione, sostituire l'endpoint del vecchio nodo con l'endpoint del nuovo nodo.
5. Se Multi-AZ è stato inizialmente disabilitato, riabilitarlo ora. Per istruzioni, consultare [Abilitazione della funzione Multi-AZ](#).

Nodi riservati ElastiCache

La prenotazione di uno o più nodi può essere un modo per ridurre i costi. I nodi riservati vengono addebitati un costo anticipato che dipende dal tipo di nodo e dalla durata della prenotazione, uno o tre anni.

Per verificare se i nodi riservati comportano un risparmio sui costi per i propri casi d'uso, determinare innanzitutto la dimensione del nodo e il numero di nodi necessari. Quindi stimare l'utilizzo del nodo e confrontare il costo totale risultante dall'utilizzo dei nodi on demand rispetto all'utilizzo dei nodi riservati. Puoi combinare l'utilizzo di nodi prenotati e on demand nei cluster. Per informazioni sui prezzi, consulta [Prezzi Amazon ElastiCache](#).

Note

I nodi riservati non sono flessibili; si applicano solo al tipo di istanza esatto che si riserva.

Gestione dei costi con nodi riservati

Prenotare uno o più nodi può essere un modo per ridurre di costi. I nodi riservati vengono addebitati un costo anticipato che dipende dal tipo di nodo e dalla durata della prenotazione, ovvero uno o tre anni. Tale addebito è molto inferiore all'addebito orario di utilizzo previsto in caso di nodi on demand.

Per verificare se i nodi riservati comportano un risparmio sui costi per i propri casi d'uso, determinare innanzitutto la dimensione del nodo e il numero di nodi necessari. Quindi stimare l'utilizzo del nodo e confrontare il costo totale risultante dall'utilizzo dei nodi on demand rispetto all'utilizzo dei nodi riservati. Puoi combinare l'utilizzo di nodi prenotati e on demand nei cluster. Per informazioni sui prezzi, consulta [Prezzi Amazon ElastiCache](#).

Regione AWS, tipo di nodo e durata devono essere scelti al momento dell'acquisto e non possono essere modificati in un secondo momento.

Puoi utilizzare AWS Management Console, AWS CLI o l'API ElastiCache per elencare e acquistare le offerte di nodi riservati disponibili.

Per ulteriori informazioni sui nodi riservati, consulta [Nodi riservati di Amazon ElastiCache](#).

Argomenti

- [Offerte di nodi riservati standard](#)
- [Offerte di nodi riservati legacy](#)

- [Recupero di informazioni sulle offerte di nodi riservati](#)
- [Acquisto di un nodo riservato](#)
- [Recupero di informazioni sui nodi riservati](#)

Offerte di nodi riservati standard

Quando acquisti un'istanza di nodi standard riservata (RI) in Amazon ElastiCache, acquisti un impegno che ti permette di ottenere una tariffa scontata per un tipo di istanza di nodo di cache e una regione AWS specifici, per la durata dell'istanza di nodo riservata. Per usare un'istanza database di nodo riservata Amazon ElastiCache, devi creare un'istanza di nodo ElastiCache con una procedura analoga a quella per la creazione di un'istanza on demand.

La nuova istanza di nodo creata deve corrispondere esattamente alle specifiche dell'istanza di nodo riservata. Se le specifiche della nuova istanza di nodo corrispondono a un'istanza di nodo riservata esistente per il tuo account, viene fatturata la tariffa scontata per l'istanza di nodo riservata. In caso contrario, l'istanza di nodo viene fatturata in base a una tariffa on demand. Le RI standard disponibili a partire dalle famiglie di istanze R5 and M5 in poi.

Note

Tutti e tre i tipi di offerta illustrati qui di seguito sono disponibili per un periodo di un anno o di tre anni.

Tipi offerta

Nessun pagamento anticipato RI permette di accedere a un'istanza ElastiCache riservata senza un pagamento anticipato. L'istanza riservata ElastiCache senza pagamento anticipato viene fatturata applicando una tariffa oraria scontata per ogni ora entro il termine, indipendentemente dall'utilizzo.

Pagamento anticipato parziale RI richiede il pagamento anticipato di una parte dell'istanza ElasticCache riservata. Le ore rimanenti del periodo di prenotazione vengono fatturate a una tariffa oraria scontata, indipendentemente dall'utilizzo. Questa opzione sostituisce l'opzione precedente **Utilizzo elevato**, illustrata nella sezione seguente.

Pagamento anticipato intero costo il servizio RI richiede il pagamento completo all'inizio del mandato del RI. Il pagamento viene effettuato per intero all'inizio del periodo della prenotazione e non vengono addebitati altri costi per il resto del periodo, indipendentemente dal numero di ore di utilizzo.

Offerte di nodi riservati legacy

Esistono tre livelli di prenotazioni dei nodi legacy: utilizzo pesante, utilizzo medio e utilizzo leggero. I nodi possono essere riservati a qualsiasi livello di utilizzo per 1 o 3 anni. Tipo di nodo, livello di utilizzo e periodo di prenotazione influiscono sui costi totali. Prima di acquistare i nodi riservati, confronta i vari modelli per verificare il risparmio che ciascun modello può garantire alla tua attività.

I nodi acquistati in base a un livello o periodo di utilizzo non possono essere convertiti in un altro livello o periodo di utilizzo.

Livelli di utilizzo

Nodi riservati con utilizzo Heavy consentono carichi di lavoro con una base di capacità coerente o eseguono carichi di lavoro con stato costante. I nodi riservati con utilizzo Heavy richiedono un elevato costo anticipato, ma se prevedi un'esecuzione per più del 79 percento del periodo contrattualizzato di nodi riservati puoi ottenere un maggiore risparmio (fino al 70 percento in meno sul prezzo on demand). I nodi riservati con utilizzo pesante prevedono un singolo addebito. Questo viene quindi seguito da una tariffa oraria più bassa per la durata del periodo, indipendentemente dal fatto che il nodo sia in esecuzione.

I nodi riservati con utilizzo Medium rappresentano l'opzione migliore se prevedi di utilizzare i nodi riservati per una notevole quantità di tempo, ma desideri un singolo addebito di minore entità o la flessibilità di interrompere il pagamento del nodo una volta spento. I nodi riservati con utilizzo Medium rappresentano l'opzione più conveniente se prevedi un'esecuzione per più del 40 percento del periodo contrattualizzato di nodi riservati. Questa opzione ti consente di risparmiare fino al 64 percento sul prezzo on demand. Con i nodi riservati con utilizzo Medium, paghi un singolo addebito leggermente superiore rispetto ai nodi riservati con utilizzo Light e disponi di tariffe orarie di utilizzo inferiori applicabili durante l'esecuzione di un nodo.

I nodi riservati con utilizzo Light sono ideali per carichi di lavoro periodici eseguiti solo un paio di ore al giorno o alcuni giorni a settimana. I nodi riservati con utilizzo Light prevedono un singolo addebito e una tariffa oraria di utilizzo scontata applicabile quando il nodo è in esecuzione. Il risparmio inizia a essere evidente quando il nodo viene eseguito per più del 17 percento del periodo contrattuale di nodi riservati. Per l'intero periodo del nodo riservato, puoi risparmiare fino al 56 percento delle tariffe on demand.

Offerte di nodi riservati legacy

Offerta	Costo anticipato	Tariffa di utilizzo	Vantaggio
---------	------------------	---------------------	-----------

Offerta	Costo anticipato	Tariffa di utilizzo	Vantaggio
Utilizzo pesante	Massimo	Tariffa oraria minore. Applicabile all'intero periodo indipendentemente dall'utilizzo del nodo riservato.	Costo complessivo inferiore se prevedi di eseguire i nodi riservati per oltre il 79 per cento di un periodo di tre anni.
Medium utilizzo	Media	Tariffa oraria di utilizzo applicata per ogni ora di esecuzione e del nodo. Nessun costo orario applicato se il nodo non viene eseguito.	Ideale per carichi di lavoro elastici o quando prevedi un utilizzo moderato, per oltre il 40 per cento di un periodo di tre anni.
Utilizzo Light	Minimo	Tariffa oraria di utilizzo applicata per ogni ora di esecuzione e del nodo. Nessun costo orario applicato se il nodo non viene eseguito. Tariffe orarie più alte rispetto a tutti i tipi di offerta, sebbene vengano applicate solo se il nodo riservato è in esecuzione.	Costo complessivo più alto se si prevede di eseguire tutto il tempo. Costo complessivo più elevato se prevedi un'esecuzione continuativa. Il costo complessivo sarà invece minimo se prevedi un utilizzo poco frequente del nodo riservato, superiore al 15 per cento di un periodo di tre anni.

Offerta	Costo anticipato	Tariffa di utilizzo	Vantaggio
Uso su richiesta (nessun nodo riservato)	Nessuno	Tariffa oraria maggiore. Applicata ogni volta che il nodo è in esecuzione.	Costo orario maggiore.

Per ulteriori informazioni, consulta [Prezzi di Amazon ElastiCache](#).

Recupero di informazioni sulle offerte di nodi riservati

Puoi ottenere informazioni sulle offerte disponibili per i nodi riservati prima di effettuarne l'acquisto.

I seguenti esempi mostrano come ottenere i prezzi e i dettagli delle offerte di nodi riservati disponibili tramite la AWS Management Console, la AWS CLI e l'API ElastiCache.

Argomenti

- [Recupero di informazioni sulle offerte di nodi riservati](#)
- [Recupero di informazioni sulle offerte di nodi riservati \(AWS CLI\)](#)
- [Recupero di informazioni sulle offerte di nodi riservati \(API ElastiCache\)](#)

Recupero di informazioni sulle offerte di nodi riservati

Per ottenere i prezzi e altre informazioni sulle offerte di cluster riservati disponibili tramite la AWS Management Console, usa la procedura seguente.

Per recuperare informazioni sulle offerte di nodi riservati disponibili

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione scegliere Nodi riservati.
3. Scegliere Purchase Reserved Nodes (Acquista nodi riservati).
4. In Engine (Motore), selezionare Redis.
5. Per determinare le offerte disponibili, effettua selezioni per le opzioni seguenti:
 - Tipo di nodo
 - Termine
 - Tipo offerta

Dopo avere effettuato le selezioni desiderate, il costo per nodo e il costo totale delle tue selezioni viene mostrato in Dettagli di prenotazione.

6. Scegliere Cancel (Annulla) per evitare di acquistare tali nodi e incorrere nel conseguente addebito dei relativi costi.

Recupero di informazioni sulle offerte di nodi riservati (AWS CLI)

Per ottenere i prezzi e altre informazioni sulle offerte di nodi riservati disponibili, digita il comando seguente al prompt dei comandi.

```
aws elasticache describe-reserved-cache-nodes-offerings
```

Questa operazione genera un output simile al seguente (formato JSON):

```
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.large",
  "Duration": 94608000,
  "FixedPrice": XXXX.X,
  "UsagePrice": X.X,
  "ProductDescription": "redis",
  "OfferingType": "All Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.X,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.xlarge",
  "Duration": 94608000,
  "FixedPrice": XXXX.X,
  "UsagePrice": X.X,
  "ProductDescription": "redis",
  "OfferingType": "Partial Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.XXX,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.large",
  "Duration": 31536000,
```

```
    "FixedPrice": X.X,  
    "UsagePrice": X.X,  
    "ProductDescription": "redis",  
    "OfferingType": "No Upfront",  
    "RecurringCharges": [  
      {  
        "RecurringChargeAmount": X.XXX,  
        "RecurringChargeFrequency": "Hourly"  
      }  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [describe-reserved-cache-nodes-offerings](#) nella documentazione di riferimento di AWS CLI.

Recupero di informazioni sulle offerte di nodi riservati (API ElastiCache)

Per ottenere i prezzi e altre informazioni sulle offerte di nodi riservati disponibili, chiama l'operazione `DescribeReservedCacheNodesOfferings`.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReservedCacheNodesOfferings  
&Version=2014-12-01  
&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>
```

Per ulteriori informazioni, consulta [DescribeReservedCacheNodesOfferings](#) nella Guida di riferimento delle API ElastiCache.

Acquisto di un nodo riservato

Gli esempi seguenti mostrano come acquistare un'offerta di nodo riservato tramite la AWS Management Console, AWS CLI e l'API ElastiCache.

Important

Le procedure negli esempi seguenti implicano l'addebito sull'account AWS di costi che non potranno essere annullati.

Argomenti

- [Acquisto di un nodo riservato \(Console\)](#)
- [Acquisto di un nodo riservato \(AWS CLI\)](#)
- [Acquisto di un nodo riservato \(API ElastiCache\)](#)

Acquisto di un nodo riservato (Console)

Questo esempio mostra come acquistare un'offerta di nodi riservati specifica, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, con ID nodo riservato myreservationID.

La seguente procedura prevede l'utilizzo della AWS Management Console per acquistare l'offerta di nodi riservati per ID offerta.

Per acquistare nodi riservati

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione scegliere il collegamento Reserved Nodes (Nodi riservati).
3. Scegliere il pulsante Purchase Reserved Nodes (Acquista nodi riservati).
4. In Engine (Motore), selezionare Redis.
5. Per determinare le offerte disponibili, effettua selezioni per le opzioni seguenti:
 - Tipo di nodo
 - Termine
 - Tipo offerta
 - Un ID nodo riservato opzionale

Dopo avere effettuato le selezioni desiderate, il costo per nodo e il costo totale delle tue selezioni vengono mostrati in Dettagli di prenotazione.

6. Scegliere Purchase (Acquista).

Acquisto di un nodo riservato (AWS CLI)

L'esempio seguente mostra come acquistare l'offerta di cluster riservati specifica, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, con ID nodo riservato myreservationID.

Al prompt dei comandi, digita il comando seguente.

Per Linux, macOS o Unix:

```
aws elasticache purchase-reserved-cache-nodes-offering \  
  --reserved-cache-nodes-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f \  
  --reserved-cache-node-id myreservationID
```

Per Windows:

```
aws elasticache purchase-reserved-cache-nodes-offering ^ \  
  --reserved-cache-nodes-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f ^ \  
  --reserved-cache-node-id myreservationID
```

Il comando restituisce un output simile al seguente:

RESERVATION	ReservationId	Class	Start Time	Duration	
Fixed Price	Usage Price	Count	State	Description	Offering Type
RESERVATION	myreservationid	cache.xx.small	2013-12-19T00:30:23.247Z	1y	
XXX.XX USD	X.XXX USD	1	payment-pending	memcached	Medium Utilization

Per ulteriori informazioni, consulta [purchase-reserved-cache-nodes-offering](#) nella documentazione di riferimento ad AWS CLI.

Acquisto di un nodo riservato (API ElastiCache)

L'esempio seguente mostra come acquistare l'offerta di nodi riservati specifica, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, con ID cluster riservato myreservationID.

Chiama l'operazione `PurchaseReservedCacheNodesOffering` con i parametri seguenti:

- ReservedCacheNodesOfferingId = 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f
- ReservedCacheNodeID = myreservationID
- CacheNodeCount = 1

Example

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=PurchaseReservedCacheNodesOffering  
  &ReservedCacheNodesOfferingId=649fd0c8-cf6d-47a0-bfa6-060f8e75e95f  
  &ReservedCacheNodeID=myreservationID  
  &CacheNodeCount=1  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Per ulteriori informazioni, consulta [PurchaseReservedCacheNodesOffering](#) nella Guida di riferimento delle API ElastiCache.

Recupero di informazioni sui nodi riservati

Puoi ottenere informazioni sui nodi riservati che hai acquistato tramite la AWS Management Console, la AWS CLI e l'API ElastiCache.

Argomenti

- [Recupero di informazioni sui nodi riservati \(Console\)](#)
- [Recupero di informazioni sui nodi riservati \(AWS CLI\)](#)
- [Recupero di informazioni sui nodi prenotati \(API Elasticache\)](#)

Recupero di informazioni sui nodi riservati (Console)

La procedura seguente illustra come utilizzare la AWS Management Console per ottenere informazioni sui nodi riservati che hai acquistato.

Per ottenere informazioni sui nodi riservati che hai acquistato

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione scegliere il collegamento Reserved Nodes (Nodi riservati).

I nodi riservati del tuo account sono presenti nell'elenco Nodi riservati. Puoi scegliere qualsiasi nodo riservato nell'elenco per visualizzare le relative informazioni dettagliate nel riquadro dei dettagli nella parte inferiore della console.

Recupero di informazioni sui nodi riservati (AWS CLI)

Per ottenere informazioni sui nodi riservati per l'account AWS, digita il seguente comando al prompt dei comandi:

```
aws elasticache describe-reserved-cache-nodes
```

Questa operazione genera un output simile al seguente (formato JSON):

```
{
  "ReservedCacheNodeId": "myreservationid",
  "ReservedCacheNodesOfferingId": "649fd0c8-cf6d-47a0-bfa6-060f8e75e95f",
  "CacheNodeType": "cache.xx.small",
  "DataTiering": "disabled",
```

```
"Duration": "31536000",
"ProductDescription": "memcached",
"OfferingType": "Medium Utilization",
"MaxRecords": 0
}
```

Per ulteriori informazioni, consulta [describe--reserved-cache-nodes](#) nella documentazione di riferimento ad AWS CLI.

Recupero di informazioni sui nodi prenotati (API Elasticache)

Per ottenere informazioni sui nodi riservati per l'account AWS in uso, chiama l'operazione `DescribeReservedCacheNodes`.

Example

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeReservedCacheNodes
&Version=2014-12-01
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

Per ulteriori informazioni, consulta [DescribeReservedCacheNodes](#) nella Guida di riferimento delle API ElastiCache.

Migrazione dei nodi della generazione precedente

I nodi di generazione precedente sono tipi di nodi che vengono eliminati gradualmente. Se non si dispone di cluster esistenti che utilizzano un tipo di nodo di generazione precedente, ElastiCache non supporta la creazione di nuovi cluster con tale tipo di nodo.

A causa della quantità limitata di tipi di nodi della generazione precedente, non è possibile garantire una sostituzione corretta quando un nodo diventa non integro nei cluster. In tale scenario, la disponibilità del cluster potrebbe essere influenzata negativamente.

Si consiglia di eseguire la migrazione dei cluster in un nuovo tipo di nodo per migliorare la disponibilità e le prestazioni. Per un tipo di nodo consigliato a cui eseguire la migrazione, vedere [Percorsi di aggiornamento](#). Per un elenco completo dei tipi di nodo supportati e dei tipi di nodo della generazione precedente in ElastiCache, vedere [Tipi di nodi supportati](#).

Migrazione dei nodi in un cluster Redis

La procedura seguente descrive come eseguire un dimensionamento di un tipo di nodo cluster Redis utilizzando la console di gestione ElastiCache. Durante questo processo, il cluster Redis continuerà ad elaborare le richieste con tempi di inattività minimi. A seconda della configurazione del cluster, è possibile che vengano visualizzati i seguenti tempi di inattività. Di seguito sono riportate stime e possono differire in base alle configurazioni specifiche:

- La modalità cluster disabilitata (nodo singolo) può vedere circa 60 secondi, principalmente a causa della propagazione DNS.
- La modalità cluster disabilitata (con nodo di replica) può vedere circa 1 secondo per i cluster che eseguono Redis 5.0.6 e versioni successive. Tutta la versione inferiore può sperimentare circa 10 secondi.
- La modalità cluster abilitata può vedere circa 1 secondo.

Per modificare un tipo di nodo cluster Redis utilizzando la console:

1. Accedere alla console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dal riquadro di navigazione scegli Cluster Redis.
3. Nell'elenco di cluster, scegliere quello da migrare.
4. Scegliere Actions (Operazioni), quindi Modify (Modifica).
5. Scegli il nuovo tipo di nodo dall'elenco dei tipi di nodo.
6. Per eseguire subito il processo di migrazione, scegliere la casella Apply immediately (Applica immediatamente). Se non viene selezionata la casella Apply immediately (Applica immediatamente), il processo di migrazione viene eseguito nel corso della successiva finestra di manutenzione del cluster.
7. Scegli Modifica. Se, nel passaggio precedente, è stato scelto Apply immediately (Applica immediatamente), lo stato del cluster diventa modifica in corso. Quando lo stato cambia in disponibile, la modifica è completa ed è possibile iniziare a utilizzare il nuovo cluster.

Per modificare un tipo di nodo del cluster Redis utilizzando il AWS CLI:

Utilizzo dell'API [modify-replication-group](#) come illustrato di seguito:

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group /  
  --replication-group-id my-replication-group /  
  --cache-node-type new-node-type /  
  --apply-immediately
```

Per Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id my-replication-group ^  
  --cache-node-type new-node-type ^  
  --apply-immediately
```

In questo scenario, il valore di *new-node-type* è il tipo di nodo a cui si sta eseguendo la migrazione. Passando il parametro `--apply-immediately`, l'aggiornamento verrà applicato immediatamente quando il gruppo di replica passa dallo stato `modifying` (modifica in corso) a `available` (disponibile). Se non viene selezionata la casella `Apply immediately` (Applica immediatamente), il processo di migrazione viene eseguito nel corso della successiva finestra di manutenzione del cluster.

Note

Se non è possibile modificare il cluster con un `InvalidCacheClusterState`, è necessario rimuovere prima un nodo di ripristino non riuscito.

Correzione o rimozione di nodi di ripristino non riusciti

La procedura seguente descrive come correggere o rimuovere i nodi di ripristino non riusciti dal cluster Redis. Per ulteriori informazioni su come i nodi ElastiCache entrano nello stato di ripristino non riuscito, consulta [Visualizzazione dello stato ElastiCache del nodo](#). Consigliamo innanzitutto di rimuovere tutti i nodi nello stato di ripristino non riuscito, quindi di migrare i nodi di generazione

precedente rimanenti nel cluster ElastiCache a un tipo di nodo di nuova generazione e infine di aggiungere nuovamente il numero di nodi richiesto.

Per rimuovere il nodo di ripristino non riuscito (console):

1. Accedere alla console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dal riquadro di navigazione scegli Cluster Redis.
3. Dall'elenco dei cluster, scegliere il nome del cluster da cui rimuovere un nodo.
4. Dall'elenco di partizioni, scegliere la shard da cui si desidera rimuovere un nodo. Ignorare questo passaggio se la modalità cluster è disattivata per il cluster.
5. Dall'elenco dei nodi, scegliere il nodo con lo stato di `restore-failed`.
6. Scegliere Actions (Operazioni), quindi Delete node (Elimina nodo).

Dopo aver rimosso i nodi di ripristino non riusciti dal cluster ElastiCache, puoi migrare a un tipo di ultima generazione. Per ulteriori informazioni, consulta la sezione precedente [Migrazione dei nodi in un cluster Redis](#).

Per aggiungere nuovamente nodi al cluster ElastiCache, consulta [Aggiunta di nodi a un cluster](#).

Gestione dei cluster

Un cluster è una raccolta di uno o più nodi di cache, ognuno dei quali esegue un'istanza del software del motore di cache Redis. Quando si crea un cluster, occorre specificare il motore e la versione che verrà utilizzata da tutti i nodi.

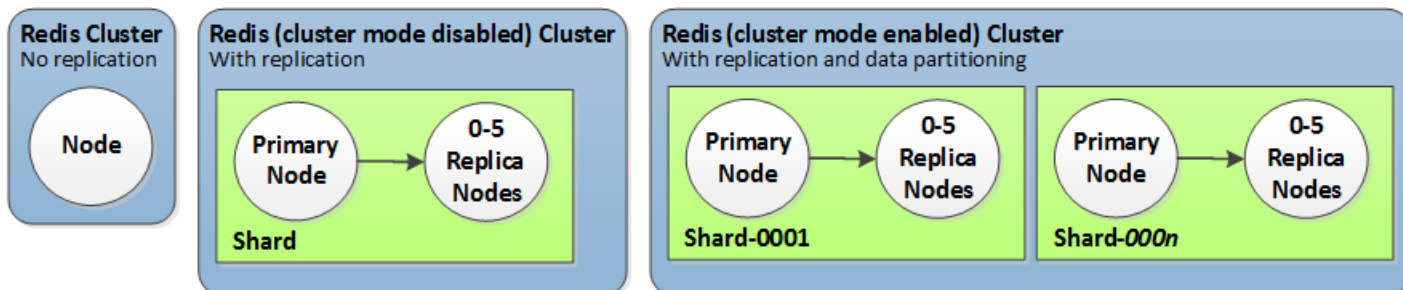
Il diagramma seguente illustra un tipico cluster Redis. I cluster Redis possono contenere un solo nodo o fino a sei nodi all'interno di una partizione (API/CLI: gruppo di nodi). Un cluster Redis a nodo singolo (modalità cluster disabilitata) è privo di partizioni e un cluster Redis a più nodi (modalità cluster disabilitata) ne ha solo una. I cluster Redis (modalità cluster abilitata) possono avere fino a 500 shard, con i dati partizionati tra le shard. Il limite del nodo o della partizione può essere aumentato fino a un massimo di 500 per cluster se la versione del motore Redis è 5.0.6 o superiore. Ad esempio, è possibile scegliere di configurare un cluster a 500 nodi che varia tra 83 partizioni (un primario e 5 repliche per partizione) e 500 partizioni (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. Per ulteriori informazioni,

consulta [Creazione di un gruppo di sottoreti](#). Per le versioni inferiori alla 5.0.6, il limite è 250 per cluster.

Per richiedere un aumento dei limiti, consulta [AWS Limiti del servizio](#) e selezionare il tipo di limite Nodi per cluster per tipo di istanza.

Quando in una partizione sono presenti più nodi, uno di essi è il nodo primario lettura/scrittura. Tutti gli altri nodi nella partizione sono repliche di sola lettura.

I tipici cluster Redis appaiono come segue.



La maggior parte delle ElastiCache operazioni 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'ElastiCache API e i comandi. AWS CLI L'identificatore del cluster deve essere univoco per quel cliente in una regione. AWS

ElastiCache supporta più versioni del motore. A meno che non ci siano motivi specifici, consigliamo di utilizzare l'ultima versione.

ElastiCache i cluster sono progettati per essere accessibili utilizzando un'istanza Amazon EC2. Se il cluster viene avviato in un virtual private cloud (VPC) basato sul servizio Amazon VPC, vi si può accedere dall'esterno di AWS. Per ulteriori informazioni, consulta [Accesso alle risorse ElastiCache dall'esterno AWS](#).

Per un elenco delle versioni Redis supportate, consulta [ElastiCache supportate per le versioni Redis](#).

Scelta del tipo di rete

ElastiCache supporta Internet Protocol versioni 4 e 6 (IPv4 e IPv6), consentendo di configurare il cluster per accettare:

- solo connessioni IPv4,
- solo connessioni IPv6,
- connessioni IPv4 e IPv6 (dual-stack)

IPv6 è supportato per i carichi di lavoro che utilizzano il motore Redis dalla versione 6.2 in avanti su tutte le istanze sviluppate sul [sistema Nitro](#). Non sono previsti costi aggiuntivi per l'accesso a ElastiCache su IPv6.

Note

La migrazione dei cluster creati prima della disponibilità di IPV6/dual-stack non è supportata. Anche il passaggio da un tipo di rete all'altro nei nuovi cluster creati non è supportato.

Configurazione di sottoreti per tipo di rete

Se crei un cluster in un Amazon VPC, devi specificare un gruppo di sottoreti. ElastiCache utilizza il gruppo di sottoreti per selezionare una sottorete e gli indirizzi IP all'interno di quella sottorete da associare ai nodi. I cluster ElastiCache richiedono una sottorete dual-stack con indirizzi IPv4 e IPv6 assegnati agli stessi per funzionare in modalità dual-stack e una sottorete solo IPv6 per funzionare come solo IPv6.

Utilizzo di dual-stack

Quando si utilizza ElastiCache per Redis in modalità cluster abilitata, dal punto di vista dell'applicazione, la connessione a tutti i nodi del cluster tramite l'endpoint di configurazione non è diversa dalla connessione diretta a un singolo nodo di cache. Per ottenere questo risultato, un client basato su cluster deve essere coinvolto in un processo di individuazione del cluster e richiedere le informazioni di configurazione per tutti i nodi. Il protocollo di individuazione di Redis supporta un solo IP per nodo.

Per mantenere la compatibilità con le versioni precedenti per tutti i client esistenti, viene introdotta l'individuazione IP, che consente di selezionare il tipo di IP (ad esempio IPv4 o IPv6) da pubblicizzare

nel protocollo di individuazione. Sebbene ciò limiti l'individuazione automatica a un solo tipo di IP, il dual-stack è comunque vantaggioso per carichi di lavoro modalità cluster abilitata, in quanto consente migrazioni (o rollback) da un tipo di individuazione IP IPv4 a uno IPv6 senza tempi di inattività.

Cluster ElastiCache dual-stack abilitati per TLS

Quando TLS è abilitato per i cluster ElastiCache, le funzioni di individuazione del cluster (`cluster slots`, `cluster shards` e `cluster nodes`) restituiscono nomi host anziché IP. I nomi host vengono quindi utilizzati al posto degli IP per connettersi al cluster ElastiCache ed eseguire un handshake TLS. Ciò significa che i client non saranno interessati dal parametro Individuazione IP. Per i cluster abilitati per TLS, il parametro Individuazione IP non ha alcun effetto sul protocollo IP preferito. Invece, il protocollo IP utilizzato verrà determinato in base a quello preferito dal client durante la risoluzione dei nomi host DNS.

Per esempi su come configurare una preferenza protocollo IP durante la risoluzione dei nomi host DNS, consulta [Cluster dual stack abilitati per TLS ElastiCache](#).

Utilizzo di AWS Management Console

Durante la creazione di un cluster mediante la AWS Management Console, in Connectivity (Connettività), scegliere un tipo di rete, IPv4, IPv6 o Dual stack. Se si sta creando un cluster Redis (modalità cluster abilitata) e si sceglie dual stack, occorre selezionare un tipo di individuazione IP: IPv6 o IPv4.

Per ulteriori informazioni, consulta [Creazione di un cluster Redis \(modalità cluster abilitata\) \(console\)](#) o [Creazione di un Redis \(modalità cluster disabilitata\) \(Console\)](#).

Durante la creazione di un gruppo di replica mediante la AWS Management Console, scegli un tipo di rete: IPv4, IPv6 o Dual stack. Se si sceglie dual-stack, è necessario selezionare un tipo di individuazione IP: IPv6 o IPv4.

Per ulteriori informazioni, consulta [Creazione di un gruppo di replica Redis \(modalità cluster disabilitata\) da zero](#) o [Creazione da zero di un gruppo di replica in Redis \(modalità cluster abilitata\)](#).

Utilizzo della CLI

Durante la creazione di un cluster di cache mediante la CLI, si utilizza il comando [create-cache-cluster](#) e si specificano i parametri `NetworkType` e `IPDiscovery`:

Per Linux, macOS o Unix:

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id "cluster-test" \  
  --engine redis \  
  --cache-node-type cache.m5.large \  
  --num-cache-nodes 1 \  
  --network-type dual_stack \  
  --ip-discovery ipv4
```

Per Windows:

```
aws elasticache create-cache-cluster ^  
  --cache-cluster-id "cluster-test" ^  
  --engine redis ^  
  --cache-node-type cache.m5.large ^  
  --num-cache-nodes 1 ^  
  --network-type dual_stack ^  
  --ip-discovery ipv4
```

Quando si crea un gruppo di replica con la modalità cluster disabilitata mediante la CLI, si utilizza il comando [create-replication-group](#) e si specificano i parametri NetworkType e IPDiscovery:

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id sample-repl-group \  
  --replication-group-description "demo cluster with replicas" \  
  --num-cache-clusters 3 \  
  --primary-cluster-id redis01 \  
  --network-type dual_stack \  
  --ip-discovery ipv4
```

Per Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --replication-group-description "demo cluster with replicas" ^  
  --num-cache-clusters 3 ^  
  --primary-cluster-id redis01 ^
```

```
--network-type dual_stack ^
--ip-discovery ipv4
```

Quando si crea un gruppo di replica con la modalità cluster abilitata e si utilizza IPv4 per individuazione IP mediante la CLI, si utilizza il comando [create-replication-group](#) e si specificano i parametri `NetworkType` e `IPDiscovery`:

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group \
  --replication-group-id demo-cluster \
  --replication-group-description "demo cluster" \
  --cache-node-type cache.m5.large \
  --num-node-groups 2 \
  --engine redis \
  --cache-subnet-group-name xyz \
  --network-type dual_stack \
  --ip-discovery ipv4 \
  --region us-east-1
```

Per Windows:

```
aws elasticache create-replication-group ^
  --replication-group-id demo-cluster ^
  --replication-group-description "demo cluster" ^
  --cache-node-type cache.m5.large ^
  --num-node-groups 2 ^
  --engine redis ^
  --cache-subnet-group-name xyz ^
  --network-type dual_stack ^
  --ip-discovery ipv4 ^
  --region us-east-1
```

Quando si crea un gruppo di replica con la modalità cluster abilitata e si utilizza IPv6 per individuazione IP mediante la CLI, si utilizza il comando [create-replication-group](#) e si specificano i parametri `NetworkType` e `IPDiscovery`:

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group \
  --replication-group-id demo-cluster \
  --replication-group-description "demo cluster" \
```

```
--cache-node-type cache.m5.large \  
--num-node-groups 2 \  
--engine redis \  
--cache-subnet-group-name xyz \  
--network-type dual_stack \  
--ip-discovery ipv6 \  
--region us-east-1
```

Per Windows:

```
aws elasticache create-replication-group ^  
--replication-group-id demo-cluster ^  
--replication-group-description "demo cluster" ^  
--cache-node-type cache.m5.large ^  
--num-node-groups 2 ^  
--engine redis ^  
--cache-subnet-group-name xyz ^  
--network-type dual_stack ^  
--ip-discovery ipv6 ^  
--region us-east-1
```

Tiering di dati

I cluster che comprendono un gruppo di replica e utilizzano un tipo di nodo della famiglia r6gd hanno i loro dati suddivisi 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. 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 aggiuntiva quando si accede ai dati su SSD.

Nei cluster con tiering dei dati, ElastiCache monitora l'ultimo orario di accesso di ogni elemento archiviato. Quando la memoria disponibile (DRAM) è completamente consumata, ElastiCache utilizza un algoritmo utilizzato più recentemente (LRU) per spostare automaticamente gli elementi a cui si accede meno frequentemente dalla memoria all'SSD. Quando successivamente si accede ai dati sull'SSD, li riporta ElastiCache automaticamente e in modo asincrono in 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 di 500 byte, è possibile prevedere in media ulteriori 300 microsecondi di latenza per le richieste ai dati archiviati su SSD rispetto alle richieste ai dati in memoria.

Con le dimensioni più grandi dei nodi di tiering di dati (cache.r6gd.16xlarge), è possibile archiviare fino a 1 petabyte in un singolo cluster a 500 nodi (500 TB quando si utilizza 1 replica di lettura). Il tiering dei dati è compatibile con tutti i comandi e le strutture dati Redis supportati in ElastiCache. Non è necessaria alcuna modifica lato client per utilizzare questa caratteristica.

Argomenti

- [Best practice](#)
- [Limitazioni](#)
- [Prezzi](#)
- [Monitoraggio](#)
- [Utilizzo del tiering di dati](#)
- [Ripristino di dati da backup su cluster con tiering di dati abilitato](#)

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. Quando gli elementi vengono spostati tra DRAM e SSD, le chiavi rimarranno sempre in memoria e solo i valori verranno spostati al livello SSD.

Limitazioni

Il livello di dati presenta le seguenti limitazioni:

- È possibile utilizzare il tiering di dati solo su cluster che fanno parte di un gruppo di replica.
- 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-central-1, eu-north-1, eu-west-3, ap-northeast-1, ap-southeast-1, ap-southeast-2, ap-south-1, ca-central-1 e sa-east-1.
- Il motore deve essere Redis 6.2 o versioni successive.
- Non è possibile ripristinare un backup di un cluster r6gd in un altro cluster a meno che non utilizzi anche r6gd.
- Non è possibile esportare un backup su Amazon S3 per cluster con tiering di dati.
- La migrazione online non è supportata per i cluster in esecuzione sul tipo di nodo r6gd.
- 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). Per ulteriori informazioni, consulta [Scalabilità per Redis ElastiCache](#).
- Il dimensionamento automatico è supportato sui cluster che utilizzano il tiering dei dati per Redis 7.0.7 e versioni successive. Per ulteriori informazioni, consultare [Auto Scaling ElastiCache per cluster Redis](#).
- Il tiering di dati supporta solo policy maxmemory volatile-lru, allkeys-lru, volatile-lfu, allkeys-lfu e noeviction.
- Il salvataggio forkless è supportato per Redis 7.0.7 e versioni successive. Per ulteriori informazioni, consulta [Modalità di implementazione di sincronizzazione e backup](#).
- Gli elementi più grandi di 128 MiB non vengono spostati su SSD.

Prezzi

I nodi R6gd hanno una capacità totale 4,8 volte superiore (memoria + SSD) e possono contribuire a un risparmio di oltre il 60% quando si esegue al massimo utilizzo rispetto ai nodi R6g (solo memoria). [Per ulteriori informazioni, consulta ElastiCache la pagina dei prezzi.](#)

Monitoraggio

ElastiCache for Redis offre metriche progettate specificamente per monitorare i cluster di prestazioni che utilizzano il tiering dei dati. Per monitorare il rapporto tra elementi in DRAM rispetto a SSD, puoi utilizzare la metrica `Currltems` in [Metriche per Redis](#). Puoi calcolare la percentuale come: $(\text{Currltems con Dimension: Tier} = \text{Memoria} * 100) / (\text{Currltems senza filtro dimensionale})$. Quando

la percentuale di elementi in memoria scende al di sotto del 5%, è preferibile valutare l'aumento orizzontale per i [cluster con modalità cluster abilitata](#) o l'aumento per i [cluster con modalità cluster disabilitata](#). Per ulteriori informazioni, consulta Parametri per i cluster Redis che utilizzano il tiering di dati su [Parametri di Redis](#).

Utilizzo del tiering di dati

Utilizzo della suddivisione dei dati su più livelli utilizzando il AWS Management Console

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

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

Abilitazione della suddivisione dei dati su più livelli utilizzando il AWS CLI

Quando si crea un gruppo di replica utilizzando il AWS CLI, si utilizza il tiering dei dati selezionando un tipo di nodo dalla famiglia r6gd, ad esempio cache.r6gd.xlarge e impostando il parametro. -- data-tiering-enabled

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-enabled, l'operazione avrà esito negativo.

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id redis-dt-cluster \  
  --replication-group-description "Redis cluster with data tiering" \  
  --num-node-groups 1 \  
  --replicas-per-node-group 1 \  
  --cache-node-type cache.r6gd.xlarge \  
  --engine redis \  
  --cache-subnet-group-name default \  
  --automatic-failover-enabled \  
  --data-tiering-enabled
```

Per Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id redis-dt-cluster ^  
  --replication-group-description "Redis cluster with data tiering" ^
```

```
--num-node-groups 1 ^
--replicas-per-node-group 1 ^
--cache-node-type cache.r6gd.xlarge ^
--engine redis ^
--cache-subnet-group-name default ^
--automatic-failover-enabled ^
--data-tiering-enabled
```

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

```
{
  "ReplicationGroup": {
    "ReplicationGroupId": "redis-dt-cluster",
    "Description": "Redis cluster with data tiering",
    "Status": "creating",
    "PendingModifiedValues": {},
    "MemberClusters": [
      "redis-dt-cluster"
    ],
    "AutomaticFailover": "enabled",
    "DataTiering": "enabled",
    "SnapshotRetentionLimit": 0,
    "SnapshotWindow": "06:00-07:00",
    "ClusterEnabled": false,
    "CacheNodeType": "cache.r6gd.xlarge",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
  }
}
```

Ripristino di dati da backup su cluster con tiering di dati abilitato

È possibile ripristinare un backup in un nuovo cluster con il tiering dei dati abilitato utilizzando (Console), () o (API). AWS CLI ElastiCache Quando si crea un cluster utilizzando tipi di nodo nella famiglia r6gd, il tiering di dati è abilitato.

Ripristino di dati da backup su cluster con tiering di dati abilitato (console)

Per ripristinare un backup su un nuovo cluster con tiering di dati abilitato (console)

1. Accedi AWS Management Console e apri la ElastiCache console all'[indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).

2. Nel riquadro di navigazione, scegliere Backups (Backup).
3. Nell'elenco dei backup, scegli la casella a sinistra del nome di backup da cui desideri eseguire il ripristino.
4. Scegli Restore (Ripristina).
5. Completare la finestra di dialogo Restore Cluster (Ripristina cluster). Assicurarsi di completare tutti i campi Obbligatorie e gli eventuali altri che si desidera modificare rispetto alle impostazioni predefinite.
 1. Cluster ID (ID cluster) : Obbligatorie. Il nome del nuovo cluster.
 2. Cluster mode enabled (scale out) (Modalità cluster abilitata (dimensionamento orizzontale)) : scegliere per un cluster Redis.
 3. Tipo di nodo – Specificare cache.r6gd.xlarge o qualsiasi altro tipo di nodo della famiglia r6gd.
 4. Number of Shards (Numero di partizioni) - Scegliere il numero di partizioni desiderato nel nuovo cluster (API/CLI: gruppi di nodi).
 5. Replicas per Shard (Repliche per partizioni) - Scegliere il numero di nodi di replica di lettura desiderato in ogni partizioni.
 6. Slots and keyspaces (Slot e keyspaces) - Scegliere la modalità di distribuzione delle chiavi tra le partizioni. Se si sceglie di specificare le distribuzioni chiave, completare la tabella specificando gli intervalli chiave per ogni partizioni.
 7. Availability zone(s) (Zone di disponibilità) - Specificare la modalità di selezione delle zone di disponibilità del cluster.
 8. Porta: modifica solo se il nuovo cluster deve utilizzare una porta diversa.
 9. Choose a VPC (scegliere un VPC) : Scegliere il VPC in cui creare questo cluster.
 10. Parameter Group (Gruppo di parametri) - Scegliere il gruppo di parametri che riserva memoria sufficiente per overhead Redis per il tipo di nodo selezionato.
6. Dopo aver selezionato tutte le impostazioni desiderate, scegli Crea.

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

Ripristino di dati da backup su cluster con tiering di dati abilitato (AWS CLI)

Quando si crea un gruppo di replica utilizzando AWS CLI, per impostazione predefinita viene utilizzato il tiering dei dati su più livelli selezionando un tipo di nodo dalla famiglia r6gd, ad esempio cache.r6gd.xlarge e impostando il parametro. `--data-tiering-enabled`

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-enabled`, l'operazione avrà esito negativo.

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id redis-dt-cluster \  
  --replication-group-description "Redis cluster with data tiering" \  
  --num-node-groups 1 \  
  --replicas-per-node-group 1 \  
  --cache-node-type cache.r6gd.xlarge \  
  --engine redis \  
  --cache-subnet-group-name default \  
  --automatic-failover-enabled \  
  --data-tiering-enabled \  
  --snapshot-name my-snapshot
```

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group ^\  
  --replication-group-id redis-dt-cluster ^\  
  --replication-group-description "Redis cluster with data tiering" ^\  
  --num-node-groups 1 ^\  
  --replicas-per-node-group 1 ^\  
  --cache-node-type cache.r6gd.xlarge ^\  
  --engine redis ^\  
  --cache-subnet-group-name default ^\  
  --automatic-failover-enabled ^\  
  --data-tiering-enabled ^\  
  --snapshot-name my-snapshot
```

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

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "redis-dt-cluster",  
    "Description": "Redis cluster with data tiering",  
    "Status": "creating",  
    "PendingModifiedValues": {},  
    "MemberClusters": [  
      "redis-dt-cluster"  
    ],  
  },  
}
```

```
    "AutomaticFailover": "enabled",
    "DataTiering": "enabled",
    "SnapshotRetentionLimit": 0,
    "SnapshotWindow": "06:00-07:00",
    "ClusterEnabled": false,
    "CacheNodeType": "cache.r6gd.xlarge",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
  }
}
```

Preparazione di un cluster

Di seguito troverai istruzioni per creare un cluster tramite la console ElastiCache, la AWS CLI o l'API ElastiCache.

Puoi anche creare un cluster ElastiCache mediante [AWS CloudFormation](#). Per ulteriori informazioni, consulta [AWS::ElastiCache::CacheCluster](#) nella AWS Guida per l'utente per la formazione cloud, che include una guida all'implementazione di tale approccio.

È consigliabile eseguire delle operazioni preliminari a ogni creazione di un cluster o di un gruppo di replica , in modo da non dover effettuare aggiornamenti o modifiche sin dai primi momenti.

Argomenti

- [Determina i tuoi requisiti](#)
- [Scelta delle dimensioni dei nodi](#)

Determina i tuoi requisiti

Preparazione

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

- Che tipo di istanza dei nodi ti occorre?

Per assistenza nella scelta di un tipo di nodo dell'istanza, consulta [Scelta delle dimensioni dei nodi](#).

- Intendi avviare il cluster in un cloud privato virtuale (VPC) basato su Amazon VPC?

⚠ Important

Se intendi avviare il cluster in un VPC, prima di iniziarne la creazione devi creare un gruppo di sottoreti nello stesso VPC. Per ulteriori informazioni, consulta [Sottoreti e gruppi di sottoreti](#).

ElastiCache è progettato per essere accessibile dall'interno AWS tramite Amazon EC2. Tuttavia, se il cluster viene avviato in un VPC basato su Amazon VPC e si trova in un VPC, è possibile fornire l'accesso dall'esterno di AWS. Per ulteriori informazioni, consulta [Accesso alle risorse ElastiCache dall'esterno AWS](#).

- 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](#).

Se stai eseguendo Redis, considera l'impostazione `reserved-memory` o `reserved-memory-percent`. Per ulteriori informazioni, consulta [Gestione della memoria prenotata](#).

- Devi creare un tuo 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

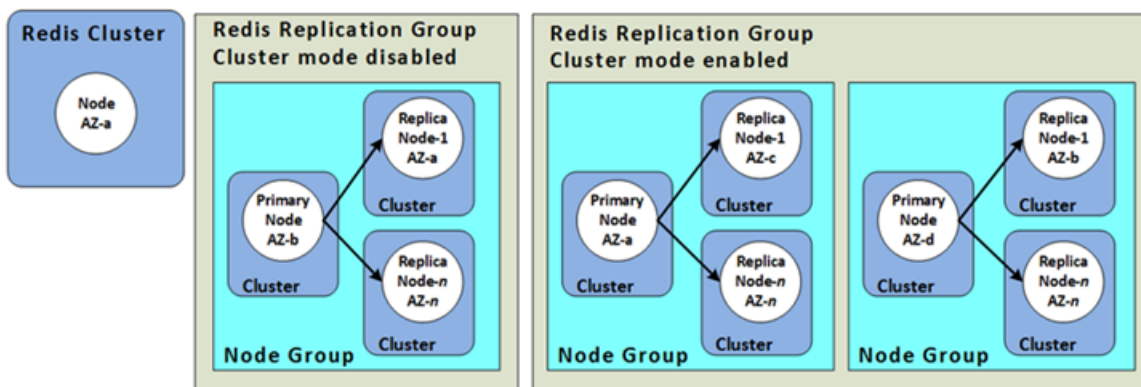
- [Requisiti di memoria e del processore](#)
- [Configurazione del cluster Redis](#)
- [Requisiti di dimensionamento](#)
- [Requisiti di accesso](#)
- [Requisiti delle Regioni, delle Zone di disponibilità e delle Local Zones](#)

Requisiti di memoria e del processore

L'elemento costitutivo di base di Amazon ElastiCache è il nodo. Per formare i cluster, i nodi vengono configurati singolarmente o a gruppi. 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 Redis

ElastiCache i cluster for Redis sono composti da 0 a 500 shard (chiamati anche gruppi di nodi). I dati presenti in un cluster Redis vengono partizionati negli shard del cluster. La tua applicazione si connette a un cluster Redis tramite un indirizzo di rete chiamato endpoint. I nodi di uno shard Redis possono svolgere due ruoli: uno è quello primario di lettura/scrittura, tutti gli altri sono nodi secondari di sola lettura (chiamati anche repliche di lettura). Oltre agli endpoint sul nodo primario, lo stesso cluster Redis ha un endpoint denominato endpoint di configurazione. L'applicazione può utilizzare questo endpoint per leggere o scrivere nel cluster, lasciando a Redis la determinazione del nodo da cui leggere o su cui scrivere. ElastiCache



Per ulteriori informazioni, consulta [Gestione dei cluster](#).

Requisiti di dimensionamento

Tutti i cluster sono dimensionabili tramite la creazione di un nuovo cluster con un nuovo e più grande tipo di nodo. Quando aumenti la dimensione di un cluster Redis, puoi avviarlo da un backup ed evitare che il nuovo cluster inizi vuoto.

Per ulteriori informazioni sul tagging, consulta [Scalabilità per Redis ElastiCache](#) in questa guida.

Requisiti di accesso

In base alla progettazione, è possibile accedere ElastiCache ai cluster Amazon dalle istanze Amazon EC2. L'accesso di rete a un ElastiCache cluster è limitato all'account che ha creato il cluster. Di conseguenza, prima di poter accedere a un cluster da un'istanza Amazon EC2, devi autorizzare l'istanza Amazon EC2 ad accedere al cluster. I passaggi per eseguire questa operazione cambiano a seconda se l'avvio è avvenuto in EC2-VPC o in EC2-Classical.

Se hai avviato il cluster in EC2-VPC devi concedere l'ingresso di rete al cluster. Se hai avviato il tuo cluster in EC2-Classical, devi concedere al gruppo di sicurezza Amazon Elastic Compute Cloud

associato all'istanza l'accesso al tuo ElastiCache gruppo di sicurezza. Per istruzioni dettagliate, consultare [Fase 3: autorizzazione dell'accesso al cluster](#) in questa guida.

Requisiti delle Regioni, delle Zone di disponibilità e delle Local Zones

Amazon ElastiCache supporta tutte le AWS regioni. Localizzando ElastiCache i cluster in una AWS regione vicina all'applicazione, puoi ridurre la latenza. Se il cluster dispone di più nodi, posizionarli in Zone di disponibilità o in Local Zones diverse può ridurre l'effetto degli errori sul cluster.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Scelta di regioni e zone di disponibilità](#)
- [Utilizzo di local zones con ElastiCache](#)
- [Limitazione dell'impatto degli errori](#)

Scelta delle dimensioni dei nodi

La scelta delle dimensioni dei nodi per un cluster influisce sui costi, le prestazioni e la tolleranza ai guasti.

Scelta delle dimensioni dei nodi

Per informazioni sui vantaggi dei processori Graviton, vedere [Processore AWS Graviton](#).

Rispondere alle seguenti domande ti può permettere di determinare il tipo di nodi al minimo necessario per l'implementazione di Redis.

- Si prevedono carichi di lavoro legati al velocità di trasmissione effettiva con più connessioni client?

Se questo è il caso e si esegue Redis 5.0.6 o versioni successive, è possibile ottenere una velocità di trasmissione effettiva e una latenza migliori utilizzando la funzionalità I/O avanzato, in cui le CPU disponibili vengono utilizzate per eseguire l'offload delle connessioni client, per conto del motore Redis. Se si esegue Redis 7.0.4 o versioni successive, oltre all'I/O avanzato, si otterrà un'accelerazione aggiuntiva con il multiplexing I/O avanzato, in cui ogni thread di I/O di rete dedicato invia i comandi da più client al motore Redis, sfruttando la capacità di Redis di elaborare in modo efficiente i comandi in batch. In ElastiCache per Redis v7.1 e versioni successive, abbiamo esteso la funzionalità dei thread di I/O avanzato per gestire anche la logica del livello di presentazione. Per livello di presentazione, intendiamo i thread di I/O avanzato che non solo leggono l'input del client, ma lo analizzano nel formato del comando binario Redis, che viene quindi

inoltrato al thread principale per l'esecuzione, fornendo un miglioramento delle prestazioni. Per ulteriori dettagli, consulta il [post del blog](#) e la pagina delle [versioni supportate](#).

- Si dispone di carichi di lavoro che accedono regolarmente a una piccola percentuale dei dati?

In questo caso e se si sta eseguendo su motore Redis versione 6.2 o successiva, è possibile sfruttare il tiering di dati scegliendo il tipo di nodo r6gd. Con il tiering di dati, i dati utilizzati meno di recente vengono archiviati su SSD. Quando vengono recuperati, è previsto un piccolo costo di latenza, bilanciato dai risparmi sui costi. Per ulteriori informazioni, consulta [Tiering di dati](#).

Per ulteriori informazioni, consulta [Tipi di nodi supportati](#).

- Di quanta memoria in totale necessiti per i tuoi dati?

Per ottenere una stima generale, prendere la dimensione degli elementi che si desidera memorizzare nella cache. Moltiplicare questa dimensione per il numero degli elementi da conservare nella cache contemporaneamente. Per ottenere una stima veritiera delle dimensioni degli elementi, serializza gli elementi della cache e contane i caratteri. Poi dividi questo numero per il numero delle partizioni nel cluster.

Per ulteriori informazioni, consulta [Tipi di nodi supportati](#).

- Quale versione di Redis stai utilizzando?

Le versioni Redis precedenti alla 2.8.22 necessitano di più memoria, per operazioni quali il failover, la snapshot, la sincronizzazione e la promozione di una replica a primaria. Occorre infatti disporre di una quantità di memoria sufficiente per tutte le operazioni di scrittura necessarie, affinché il processo vada a buon fine.

La versione 2.8.22 e quelle successive di Redis utilizzano un processo di salvataggio senza fork che richiede meno memoria disponibile rispetto al passato.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Modalità di implementazione di sincronizzazione e backup](#)
- [Verifica della disponibilità di memoria sufficiente per creare uno snapshot Redis](#)
- Quant'è consistente il carico di lavoro in scrittura della tua applicazione?

Le applicazioni caratterizzate da elevata attività di scrittura possono richiedere molta più memoria disponibile, non occupata dai dati, per acquisire snapshot o per il failover. Ogni volta che viene svolto il processo BGSAVE, è necessario disporre di memoria sufficiente che non viene utilizzata dai dati per contenere tutte le scritture che avvengono durante il processo BGSAVE. Esempi sono

quando si esegue uno snapshot, quando si sincronizza un cluster primario con una replica in un cluster e quando si abilita la caratteristica AOF (Append-only File). Un altro è la promozione di una replica al ruolo di nodo primario (se la funzione Multi-AZ è abilitata). Il caso peggiore è quando tutti i tuoi dati vengono riscritti durante il processo. In questo caso, è necessaria una dimensione di istanza di nodo con il doppio della memoria necessaria per i soli dati.

Per informazioni più dettagliate, consulta [Verifica della disponibilità di memoria sufficiente per creare uno snapshot Redis](#).

- La tua implementazione sarà un cluster Redis (modalità cluster disabilitata) o un cluster Redis (modalità cluster abilitata) con più partizioni?

Cluster Redis (modalità cluster disattivata)

Per implementare un cluster Redis (modalità cluster disattivata), il tipo di nodo deve essere in grado di ospitare tutti i dati più il sovraccarico necessario, come accennato al punto precedente.

Si supponga, ad esempio, di stimare che la dimensione totale di tutti gli articoli sia di 12 GB. In questo caso, puoi usare un cache `.m3.xlarge` nodo con 13,3 GB di memoria o un cache `.r3.large` con 13,5 GB di memoria. Tuttavia, potrebbe occorrere più memoria per le operazioni di BGSAVE. Se l'applicazione è pesante da scrivere, raddoppiare i requisiti di memoria fino ad almeno 24 GB. Quindi, usa un cache `.m3.2xlarge` con 27,9 GB di memoria o cache `.r3.xlarge` con 30,5 GB di memoria.

Redis (modalità cluster abilitata) con più partizioni

Per implementare un cluster Redis con più partizioni, il tipo di nodo deve essere in grado di ospitare i byte di dati `bytes-for-data-and-overhead / number-of-shards`.

Se, ad esempio, stimi in 12 GB la dimensione totale di tutti gli elementi e disponi di due partizioni. In questo caso, puoi usare un nodo cache `.m3.large` con 6,05 GB di memoria (12 GB/2). Tuttavia, potrebbe occorrere più memoria per le operazioni di BGSAVE. Se l'applicazione è pesante da scrivere, raddoppiare i requisiti di memoria fino ad almeno 12 GB per partizioni. Quindi, usa un cache `.m3.xlarge` con 13,3 GB di memoria o cache `.r3.large` con 13,5 GB di memoria.

- Stai usando Local Zones?

[Local Zones](#) consente di posizionare risorse come un cluster ElastiCache in più posizioni vicine agli utenti. Tuttavia, quando si sceglie la dimensione del nodo, tenere presente che le dimensioni dei nodi disponibili sono limitate alle seguenti al momento, indipendentemente dai requisiti di capacità:

- **Generazione attuale:**

Tipi di nodi M5: `cache.m5.large`, `cache.m5.xlarge`, `cache.m5.2xlarge`,
`cache.m5.4xlarge`, `cache.m5.12xlarge`, `cache.m5.24xlarge`

Tipi di nodi R5: `cache.r5.large`, `cache.r5.xlarge`, `cache.r5.2xlarge`,
`cache.r5.4xlarge`, `cache.r5.12xlarge`, `cache.r5.24xlarge`

Tipi di nodo T3: `cache.t3.micro`, `cache.t3.small`, `cache.t3.medium`

Mentre il cluster è in esecuzione, puoi monitorare l'utilizzo della memoria e del processore, gli hit della cache e i parametri dei mancati riscontri nella cache, pubblicati su CloudWatch. È possibile notare che il cluster non ha la frequenza di accessi desiderata o che le chiavi vengono espulse troppo spesso. In questi casi, puoi scegliere una dimensione di nodi diversa con specifiche di memoria e CPU maggiori.

Durante il monitoraggio dell'utilizzo della CPU, ricordare che Redis è single-threaded. Pertanto, moltiplica l'utilizzo della CPU segnalato per il numero di core CPU per ottenere l'effettivo utilizzo. Ad esempio, una CPU a quattro core che segnala un tasso di utilizzo del 20% corrisponde, in realtà, a un unico core utilizzato da Redis all'80%.

Creazione di un cluster

Gli esempi seguenti mostrano come creare un cluster Redis utilizzando l' AWS Management Console ElastiCache API AWS CLI e.

Creazione di un Redis (modalità cluster disabilitata) (Console)

ElastiCache supporta la replica quando si utilizza il motore Redis. Per monitorare la latenza tra il momento in cui i dati vengono scritti su un cluster primario di lettura/scrittura Redis e il momento in cui vengono propagati in un cluster secondario di sola lettura, aggiunge al cluster una chiave speciale, `ElastiCacheMasterReplicationTimestamp`. Questa chiave è l'ora UTC (Universal Time Coordinated) corrente. Dato che un cluster Redis potrebbe essere aggiunto a un gruppo di replica in un secondo momento, questa chiave è inclusa in tutti i cluster Redis, anche se all'inizio non fanno parte di un gruppo di replica. Per ulteriori informazioni sui gruppi di replica, consulta [Alta disponibilità utilizzando gruppi di replica](#).

Per creare un cluster Redis (modalità cluster disabilitata), segui la procedura descritta in [Creazione di un cluster Redis \(modalità cluster disabilitata\) \(Console\)](#).

Quando lo stato del cluster diventa disponibile, è possibile concedere a Amazon EC2 l'accesso al cluster, connettersi al cluster e iniziare a utilizzarlo. Per ulteriori informazioni, consultare [Fase 3: autorizzazione dell'accesso al cluster](#) e [Fase 4: connessione a un nodo del 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 [Eliminazione di un cluster](#).

Creazione di un cluster Redis (modalità cluster abilitata) (console).

Se si utilizza Redis 3.2.4 o versioni successive, è possibile creare un cluster Redis (modalità cluster abilitata). I cluster Redis (modalità cluster abilitata) supportano il partizionamento dei dati da 1 a 500 partizioni (API/CLI: gruppi di nodi) ma con alcune limitazioni. Per un confronto tra Redis (modalità cluster disabilitata) e Redis (modalità cluster abilitata), vedere [ElastiCache supportate per le versioni Redis](#).

Per creare un cluster Redis (modalità cluster abilitata) utilizzando la console ElastiCache

1. Accedi AWS Management Console e apri la ElastiCache console Amazon all'[indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Dall'elenco nell'angolo in alto a destra, scegli la AWS regione in cui desideri avviare questo cluster.
3. Scegli Nozioni di base dal riquadro di navigazione.
4. Scegli Crea VPC e segui la procedura descritta in [Creazione di un cloud privato virtuale \(VPC\)](#).
5. Nella pagina della ElastiCache dashboard, scegli Crea cluster, quindi scegli Crea cluster Redis.
6. In Cluster settings (Impostazioni cluster), procedi come indicato di seguito:
 - a. Scegli Configurazione e creazione di un nuovo cluster.
 - b. Per Modalità cluster, scegli Abilitata.
 - c. Per Informazioni sul cluster, inserisci un valore per Nome.
 - d. (Facoltativo) Inserisci un valore per Descrizione.
7. In Location (Ubicazione):

AWS Cloud

1. Per CloudAWS , è preferibile accettare le impostazioni predefinite per Multi-AZ e Failover automatico. Per ulteriori informazioni, consulta [Ridurre al minimo i tempi di inattività in ElastiCache Redis](#) con Multi-AZ.
2. In Cluster settings (Impostazioni cluster)
 - a. Per Engine version (Versione motore), scegli una versione disponibile.
 - b. Per Porta utilizza la porta predefinita, 6379. Se esiste un motivo per utilizzare una porta diversa, immettere il numero di porta.
 - c. Per Parameter group (Gruppo di parametri), scegliere un gruppo di parametri o crearne uno nuovo. I gruppi di parametri controllano i parametri di runtime del cluster. Per ulteriori informazioni sui gruppi di parametri, consulta [Parametri specifici di Redis](#) e [Creazione di un gruppo di parametri](#).

Note

Quando si seleziona un gruppo di parametri per impostare i valori di configurazione del motore, tale gruppo di parametri viene applicato a tutti

i cluster nel Global Datastore. Nella pagina Parameter Groups (Gruppi di parametri) con/senza l'attributo Global (Globale) indica se un gruppo di parametri fa parte di un Global Datastore.

- d. In Node type (Tipo di nodo), scegliere la freccia in giù (▼).

Nella finestra di dialogo Change node type (Cambia tipo di nodo) scegliere un valore per Instance family (Famiglia di istanze) per il tipo di nodo desiderato. Scegliere quindi il tipo di nodo che si desidera utilizzare per il cluster, quindi scegliere Save (Salva).

Per ulteriori informazioni, consulta [Scelta delle dimensioni dei nodi](#).

Se scegli un tipo di nodo r6gd, il tiering dei dati viene abilitato automaticamente. Per ulteriori informazioni, consulta [Tiering di dati](#).

- e. In Number of Shards (Numero di partizioni), scegliere il numero di partizioni (partizioni/gruppi di nodi) per questo cluster Redis (modalità cluster abilitata).

Per alcune versioni di Redis (modalità cluster abilitata), è possibile modificare dinamicamente il numero dle partizioni nel cluster:

- Redis 3.2.10 e versioni successive – Se il cluster esegue Redis 3.2.10 o versioni successive, è possibile modificare dinamicamente il numero dle partizioni nel cluster. Per ulteriori informazioni, consulta [Ridimensionamento dei cluster in Redis \(modalità cluster attivata\)](#).
 - Altre versioni di Redis – Se il cluster esegue una versione di Redis precedente alla versione 3.2.10, esiste un altro approccio. Per modificare il numero dle partizioni nel cluster in questo caso, creare un nuovo cluster con il nuovo numero dle partizioni. Per ulteriori informazioni, consulta [Ripristino da un backup in una nuova cache](#).
- f. In Replicas per shard (Repliche per partizione): scegliere il numero di nodi di replica di lettura per ognle partizioni.

Esistono le seguenti restrizioni per Redis (modalità cluster abilitata).

- Se hai abilitato la funzione Multi-AZ, assicurati di avere almeno una replica per ognle partizioni.

- Quando utilizzi la console per creare il cluster, il numero delle repliche è lo stesso per ogni partizione.
- Il numero delle repliche di lettura per ogni partizione è fisso e non modificabile. Se scopri che ti servono più o meno repliche per partizione (API/CLI: gruppo di nodi), devi creare un nuovo cluster con il nuovo numero di repliche. Per ulteriori informazioni, consulta [Inizializzazione di un nuovo cluster progettato autonomamente con un backup creato esternamente](#).

3. In Connectivity (Connettività)

- a. Per Network type (Tipo di rete), scegli le versioni IP supportate da questo cluster.
- b. Per i gruppi di sottoreti, scegli la sottorete che desideri applicare a questo cluster. ElastiCache utilizza quel gruppo di sottorete per scegliere una sottorete e gli indirizzi IP all'interno di quella sottorete da associare ai nodi. ElastiCache i cluster richiedono una sottorete dual-stack con indirizzi IPv4 e IPv6 assegnati per funzionare in modalità dual-stack e una sottorete solo IPv6 per funzionare solo come IPv6.

Durante la creazione di un nuovo gruppo di sottoreti, immettere l'VPC ID (ID VPC) a cui appartiene.

Seleziona un Discovery IP type (Tipo di rilevamento IP). Vengono restituiti solo gli indirizzi IP del protocollo scelto.

Per ulteriori informazioni, consultare:

- [Scelta del tipo di rete](#).
- Creazione di una sottorete nel VPC.

Se si esegue per [Utilizzo di local zones con ElastiCache](#), è necessario creare o scegliere una sottorete che si trova nella zona locale.

Per ulteriori informazioni, consulta [Sottoreti e gruppi di sottoreti](#).


4. Per Posizionamento zone disponibilità, le opzioni sono due:

- ElastiCache Nessuna preferenza: sceglie la zona di disponibilità.
- Specify Availability Zones (Specifica zone di disponibilità) – L'utente specifica la zona di disponibilità per ogni cluster.

Se hai scelto di specificare le zone di disponibilità, per ogni cluster in ognle partizioni, seleziona la zona di disponibilità dall'elenco.


Per ulteriori informazioni, consulta [Scelta di regioni e zone di disponibilità](#).

5. Seleziona Successivo.
6. In Impostazioni avanzate di Redis
 - Per Sicurezza:
 - i. 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 scegliendo la chiave AWS KMS gestita dal cliente e scegliendo la chiave. Per ulteriori informazioni, vedere [Utilizzo di chiavi gestite dal cliente da AWS KMS](#).

- Crittografia dei dati in transito – Consente la crittografia dei dati in trasferimento. Per maggiori informazioni, consultare [Crittografia dei dati in transito](#). Per il motore Redis versione 6.0 e successive, se si abilita la Crittografia in transito, viene richiesto di specificare una delle seguenti opzioni di Controllo degli accessi:
 - Nessun controllo degli accessi - Questa è l'impostazione di default. Questo non indica alcuna restrizione sull'accesso dell'utente al cluster.
 - Lista di controllo accessi – scegliere un gruppo di utenti con un set definito di utenti che possono accedere al cluster. Per ulteriori informazioni, consulta [Gestione dei gruppi di utenti con Console e CLI](#).
 - Redis AUTH Default User (Utente di default Redis AUTH) : Un meccanismo di autenticazione per il server Redis. Per ulteriori informazioni, vedere [Redis AUTH](#).
 - Redis AUTH - Un meccanismo di autenticazione per il server Redis. Per ulteriori informazioni, vedere [Redis AUTH](#).

 Note

Per le versioni di Redis a partire da 3.2.6 e successive, esclusa la versione 3.2.10, Redis AUTH è l'unica opzione.

- ii. 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.

7. Se si desiderano backup regolari pianificati regolarmente, scegliere Enable automatic backups (Abilita backup automatici), quindi digitare il numero di giorni in cui conservare ciascun backup automatico prima che venga automaticamente eliminato. Se non si desiderano backup automatici pianificati regolarmente, deselezionare la casella di controllo Enable automatic backups (Abilita backup automatici). In ogni caso, hai sempre la possibilità di creare backup manuali.

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

8. (Facoltativo) Specifica una finestra di manutenzione. La Maintenance window (Finestra di manutenzione) è l'intervallo di tempo, generalmente un'ora a settimana, in cui ElastiCache pianifica la manutenzione di sistema per il cluster. Si può consentire a ElastiCache di scegliere l'ora e il giorno della finestra di manutenzione (No preference (Nessuna preferenza)) o sceglierne autonomamente il giorno, l'ora e la durata (Specify maintenance window (Specifica 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](#).

9. (Facoltativo) PerRegistro:
 - In Formato del registro, scegliere Text o JSON.
 - In Tipo di destinazione, scegli CloudWatch Logs o Kinesis Firehose.

- In Destinazione log, scegliete Crea nuovo e inserite il nome del gruppo di log CloudWatch Logs o il nome dello stream Firehose, oppure scegliete Seleziona esistente e quindi scegliete il nome del gruppo di log Logs o CloudWatch il nome dello stream Firehose,
10. Per quanto riguarda i tag, per aiutarti a gestire i tuoi cluster e altre ElastiCache risorse, puoi assegnare i tuoi metadati a ciascuna risorsa sotto forma di tag. Per ulteriori informazioni, consulta [Assegnazione di tag delle risorse ElastiCache](#).
 11. Seleziona Avanti.
 12. Riesaminare le voci e le selezioni, quindi apportare le eventuali correzioni. Al termine, scegli Crea.

On premises

1. Per On-premise, è preferibile lasciare abilitato il Failover automatico. Per ulteriori informazioni, consulta [Ridurre al minimo i tempi](#) di inattività in Redis con Multi-AZ ElastiCache
2. Segui la procedura descritta in [Utilizzo di outpost](#).

Per creare l'equivalente utilizzando l' ElastiCache API o al AWS CLI posto della ElastiCache console, consulta quanto segue:

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

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 [Fase 3: autorizzazione dell'accesso al cluster](#) e [Fase 4: connessione a un nodo del 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 [Eliminazione di un cluster](#).

Creazione di un cluster (AWS CLI)

Per creare un cluster utilizzando AWS CLI, utilizzare il `create-cache-cluster` comando.

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 [Eliminazione di un cluster](#).

Creazione di un cluster Redis (modalità cluster disabilitata) (CLI)

Example — Un cluster Redis (modalità cluster disabilitata) senza repliche di lettura

Il seguente codice CLI crea un cluster di cache senza repliche.

Note

Quando si crea un cluster utilizzando un tipo di nodo della famiglia r6gd, è necessario passare il parametro `data-tiering-enabled`.

Per Linux, macOS o Unix:

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--cache-node-type cache.r4.large \  
--engine redis \  
--num-cache-nodes 1 \  
--cache-parameter-group default.redis6.x \  
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

Per Windows:

```
aws elasticache create-cache-cluster ^  
--cache-cluster-id my-cluster ^  
--cache-node-type cache.r4.large ^  
--engine redis ^  
--num-cache-nodes 1 ^  
--cache-parameter-group default.redis6.x ^
```

```
--snapshot-arns arn:aws:s3:::my_bucket/snapshot.rdb
```

Creazione di un cluster Redis (modalità cluster abilitata) (AWS CLI)

I cluster Redis (modalità cluster abilitata) (API/CLI: gruppi di replica) non possono essere creati utilizzando l'`create-cache-cluster` operazione. Per creare un gruppo di replica Redis (modalità cluster abilitata) (API/CLI: gruppo di replica), utilizza [Creazione da zero di un gruppo di replica Redis \(modalità cluster abilitata\) \(AWS CLI\)](#).

Per ulteriori informazioni, vedere l'argomento AWS CLI di ElastiCache riferimento [create-replication-group](#).

Creazione di un cluster (ElastiCache API)

Per creare un cluster utilizzando l' ElastiCache API, utilizza l'`CreateCacheCluster` operazione.

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 [Eliminazione di un cluster](#).

Argomenti

- [Creazione di un cluster di cache \(ElastiCache API\) Redis \(modalità cluster disabilitata\)](#)
- [Creazione di un cluster di cache in Redis \(modalità cluster abilitata\) \(ElastiCache API\)](#)

Creazione di un cluster di cache (ElastiCache API) Redis (modalità cluster disabilitata)

Il codice seguente crea un cluster di cache (ElastiCache API) Redis (modalità cluster disabilitata).

Le interruzioni di riga vengono aggiunte per facilitare la lettura.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=CreateCacheCluster  
  &CacheClusterId=my-cluster  
  &CacheNodeType=cache.r4.large  
  &CacheParameterGroup=default.redis3.2  
  &Engine=redis  
  &EngineVersion=3.2.4
```

```
&NumCacheNodes=1
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&SnapshotArns.member.1=arn%3Aaws%3As3%3A%3A%3AmyS3Bucket%2Fdump.rdb
&Timestamp=20150508T220302Z
&Version=2015-02-02
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Credential=<credential>
&X-Amz-Date=20150508T220302Z
&X-Amz-Expires=20150508T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Signature=<signature>
```

Creazione di un cluster di cache in Redis (modalità cluster abilitata) (ElastiCache API)

I cluster Redis (modalità cluster abilitata) (API/CLI: gruppi di replica) non possono essere creati utilizzando l'`CreateCacheCluster` operazione. Per creare un gruppo di replica Redis (modalità cluster abilitata) (API/CLI: gruppo di replica), utilizza [Creazione di un gruppo di replica in Redis \(modalità cluster abilitata\) da zero \(API ElastiCache\)](#).

Per ulteriori informazioni, consulta l'argomento di riferimento sull' ElastiCache [CreateReplicationGroup](#) API.

Visualizzazione dei dettagli di un cluster

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

Visualizzazione dei dettagli di un cluster Redis (modalità cluster disabilitata) (console)

È possibile visualizzare i dettagli di un cluster Redis (modalità cluster disabilitata) utilizzando la console ElastiCache, la AWS CLI per ElasticaCache o l'API ElastiCache.

La seguente procedura mostra come visualizzare i dettagli di un cluster Redis (modalità cluster disabilitata) utilizzando la console ElastiCache.

Per visualizzare i dettagli di un cluster Redis (modalità cluster disabilitata)

1. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di controllo della console ElastiCache, scegliere Redis per visualizzare un elenco di tutti i cluster che eseguono una qualsiasi versione di Redis.
3. Per visualizzare i dettagli di un cluster, seleziona la casella di controllo a sinistra del nome del cluster. Assicurati di scegliere un cluster che esegue il motore Redis, non Redis in cluster. In questo modo vengono visualizzati i dettagli relativi al cluster, incluso l'endpoint primario del cluster.
4. Per visualizzare le informazioni del nodo:
 - a. Seleziona il nome del cluster.
 - b. Seleziona la scheda Partizioni e nodi. In questo modo vengono visualizzati i dettagli relativi a ciascun nodo, incluso l'endpoint del nodo che dovrai utilizzare per effettuare la lettura dal cluster.
5. Per visualizzare le metriche, scegli la scheda Metrics (Metriche), contenente le metriche pertinenti per tutti i nodi del cluster. Per ulteriori informazioni, consulta [Monitoraggio dell'utilizzo con i parametri di CloudWatch](#)
6. Per visualizzare i log, scegli la scheda Logs (Log), che indica se il cluster utilizza i log lenti o i log del motore e fornisce i dettagli pertinenti. Per ulteriori informazioni, consulta [Consegna dei registri](#).
7. Scegli la scheda Network and security (Rete e sicurezza) per visualizzare i dettagli sulla connettività di rete del cluster e sulla configurazione del gruppo di sottoreti. Per ulteriori informazioni, consulta [Sottoreti e gruppi di sottoreti](#).

8. Scegli la scheda Maintenance (Manutenzione) per visualizzare i dettagli sulle impostazioni di manutenzione del cluster. Per ulteriori informazioni, consulta [Gestione della manutenzione](#).
9. Scegli la scheda Service updates (Aggiornamenti del servizio) per visualizzare i dettagli sugli eventuali aggiornamenti del servizio disponibili insieme all'applicazione per data consigliata. Per ulteriori informazioni, consulta [Aggiornamenti del servizio in ElastiCache](#).
10. Scegli la scheda Tags (Tag) per visualizzare i dettagli su tutti i tag applicati alle risorse del cluster. Per ulteriori informazioni, consulta [Assegnazione di tag delle risorse ElastiCache](#).

Visualizzazione dei dettagli di un cluster Redis (modalità cluster abilitata) (console)

È possibile visualizzare i dettagli di un cluster Redis (modalità cluster abilitata) utilizzando la console ElastiCache, la AWS CLI per ElasticaCache o l'API ElastiCache.

La seguente procedura mostra come visualizzare i dettagli di un cluster utilizzando la console ElastiCache.

Per visualizzare i dettagli di un cluster Redis (modalità cluster abilitata)

1. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dall'elenco nell'angolo in alto a destra, scegliere la regione AWS a cui si è interessati.
3. Nel pannello di controllo della console ElastiCache, scegliere Redis per visualizzare un elenco di tutti i cluster che eseguono una qualsiasi versione di Redis.
4. Per visualizzare i dettagli di un cluster Redis (modalità cluster attivata), scegliere la casella a sinistra del nome del cluster. Assicurati di scegliere un cluster che esegue il motore Redis in cluster, non solo Redis.

La schermata si espande al di sotto del cluster e ne mostra i dettagli, incluso l'endpoint di configurazione.

5. Per visualizzare un elenco delle partizioni del cluster e del numero dei nodi in ogni partizione, scegli la scheda Shards and nodes (Partizioni e nodi).
6. Per visualizzare informazioni specifiche su un nodo:
 - Scegliere l'ID della partizione.

In questo modo vengono visualizzate le informazioni relative a ciascun nodo, incluso l'endpoint, da utilizzare per effettuare la lettura dei dati dal cluster.

7. Per visualizzare le metriche, scegli la scheda Metrics (Parametri), contenente le metriche pertinenti per tutti i nodi del cluster. Per ulteriori informazioni, consulta [Monitoraggio dell'utilizzo con i parametri di CloudWatch](#)
8. Per visualizzare i log, scegli la scheda Logs (Log), che indica se il cluster utilizza i log lenti o i log del motore e fornisce i dettagli pertinenti. Per ulteriori informazioni, consulta [Consegna dei registri](#).
9. Scegli la scheda Network and security (Rete e sicurezza) per visualizzare i dettagli sulla connettività di rete del cluster e sulla configurazione del gruppo di sottoreti e su quale metodo di crittografia è abilitato sul cluster. Per ulteriori informazioni, consulta [Sottoreti e gruppi di sottoreti e Sicurezza dei dati in Amazon ElastiCache](#).
10. Scegli la scheda Maintenance (Manutenzione) per visualizzare i dettagli sulle impostazioni di manutenzione del cluster. Per ulteriori informazioni, consulta [Gestione della manutenzione](#).
11. Scegli la scheda Service updates (Aggiornamenti del servizio) per visualizzare i dettagli sugli eventuali aggiornamenti del servizio disponibili insieme all'applicazione per data consigliata. Per ulteriori informazioni, consulta [Aggiornamenti del servizio in ElastiCache](#).
12. Scegli la scheda Tags (Tag) per visualizzare i dettagli su tutti i tag applicati alle risorse del cluster. Per ulteriori informazioni, consulta [Assegnazione di tag delle risorse ElastiCache](#).

Visualizzazione dei dettagli di un cluster (AWS CLI)

Il codice seguente elenca i dettagli per *my-cluster*:

```
aws elasticache describe-cache-clusters --cache-cluster-id my-cluster
```

Sostituisci *my-cluster* con il nome del cluster nel caso in cui il cluster venga creato con 1 nodo di cache e 0 shard utilizzando il comando `create-cache-cluster`.

```
{
  "CacheClusters": [
    {
      "CacheClusterStatus": "available",
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-dbe93fa2"
        }
      ]
    }
  ],
}
```

```

    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "wed:12:00-wed:13:00",
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "08:30-09:30",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false,
    "CacheClusterId": "my-cluster1",
    "CacheClusterCreateTime": "2018-02-26T21:06:43.420Z",
    "PreferredAvailabilityZone": "us-west-2c",
    "AuthTokenEnabled": false,
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis3.2"
    },
    "SnapshotRetentionLimit": 0,
    "AutoMinorVersionUpgrade": true,
    "EngineVersion": "3.2.10",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  }
}

```

```

{
  "CacheClusters": [
    {
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-dbe93fa2"
        }
      ],
      "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
      "AuthTokenEnabled": false,
      "CacheSubnetGroupName": "default",
      "SnapshotWindow": "12:30-13:30",
      "AutoMinorVersionUpgrade": true,
      "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",

```

```
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": false,
    "PreferredAvailabilityZone": "us-west-2a",
    "TransitEncryptionEnabled": false,
    "ReplicationGroupId": "my-cluster2",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
    "CacheClusterId": "my-cluster2-001",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": false,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": false,
    "PreferredAvailabilityZone": "us-west-2b",
    "TransitEncryptionEnabled": false,
    "ReplicationGroupId": "my-cluster2",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
    "CacheClusterId": "my-cluster2-002",
    "PendingModifiedValues": {},
```

```

    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "AuthTokenEnabled": false,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:13:24.250Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": false,
    "PreferredAvailabilityZone": "us-west-2c",
    "TransitEncryptionEnabled": false,
    "ReplicationGroupId": "my-cluster2",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "sun:08:30-sun:09:30",
    "CacheClusterId": "my-cluster2-003",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis3.2"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "3.2.10",

```

```

    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  }

```

```

{
  "CacheClusters": [
    {
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-dbe93fa2"
        }
      ],
      "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
      "AuthTokenEnabled": true,
      "CacheSubnetGroupName": "default",
      "SnapshotWindow": "12:30-13:30",
      "AutoMinorVersionUpgrade": true,
      "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
      "CacheClusterStatus": "available",
      "AtRestEncryptionEnabled": true,
      "PreferredAvailabilityZone": "us-west-2a",
      "TransitEncryptionEnabled": true,
      "ReplicationGroupId": "my-cluster3",
      "Engine": "redis",
      "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
      "CacheClusterId": "my-cluster3-0001-001",
      "PendingModifiedValues": {},
      "CacheNodeType": "cache.r4.large",
      "DataTiering": "disabled",
      "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "ParameterApplyStatus": "in-sync",
        "CacheParameterGroupName": "default.redis6.x.cluster.on"
      },
      "SnapshotRetentionLimit": 0,
      "EngineVersion": "6.0",
      "CacheSecurityGroups": [],
      "NumCacheNodes": 1
    },
    {
      "SecurityGroups": [

```

```
    {
      "Status": "active",
      "SecurityGroupId": "sg-dbe93fa2"
    }
  ],
  "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
  "AuthTokenEnabled": true,
  "CacheSubnetGroupName": "default",
  "SnapshotWindow": "12:30-13:30",
  "AutoMinorVersionUpgrade": true,
  "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
  "CacheClusterStatus": "available",
  "AtRestEncryptionEnabled": true,
  "PreferredAvailabilityZone": "us-west-2b",
  "TransitEncryptionEnabled": true,
  "ReplicationGroupId": "my-cluster3",
  "Engine": "redis",
  "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
  "CacheClusterId": "my-cluster3-0001-002",
  "PendingModifiedValues": {},
  "CacheNodeType": "cache.r4.large",
  "DataTiering": "disabled",
  "CacheParameterGroup": {
    "CacheNodeIdsToReboot": [],
    "ParameterApplyStatus": "in-sync",
    "CacheParameterGroupName": "default.redis3.2.cluster.on"
  },
  "SnapshotRetentionLimit": 0,
  "EngineVersion": "3.2.6",
  "CacheSecurityGroups": [],
  "NumCacheNodes": 1
},
{
  "SecurityGroups": [
    {
      "Status": "active",
      "SecurityGroupId": "sg-dbe93fa2"
    }
  ],
  "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
  "AuthTokenEnabled": true,
  "CacheSubnetGroupName": "default",
```

```

    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2c",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0001-003",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2b",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",

```



```

    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0002-001",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2c",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0002-002",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis3.2.cluster.on"
    }
  }

```

```
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "3.2.6",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  },
  {
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "AuthTokenEnabled": true,
    "CacheSubnetGroupName": "default",
    "SnapshotWindow": "12:30-13:30",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterCreateTime": "2018-02-26T21:17:01.439Z",
    "CacheClusterStatus": "available",
    "AtRestEncryptionEnabled": true,
    "PreferredAvailabilityZone": "us-west-2a",
    "TransitEncryptionEnabled": true,
    "ReplicationGroupId": "my-cluster3",
    "Engine": "redis",
    "PreferredMaintenanceWindow": "thu:11:00-thu:12:00",
    "CacheClusterId": "my-cluster3-0002-003",
    "PendingModifiedValues": {},
    "CacheNodeType": "cache.r4.large",
    "DataTiering": "disabled",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "ParameterApplyStatus": "in-sync",
      "CacheParameterGroupName": "default.redis6.x.cluster.on"
    },
    "SnapshotRetentionLimit": 0,
    "EngineVersion": "6.0",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1
  }
]
```

Nel caso in cui il cluster venga creato utilizzando AWS Management Console (nodo cluster abilitato o disabilitato con 1 o più shard), usa il seguente comando per descrivere i dettagli del cluster (sostituisci *my-cluster* con il nome del gruppo di replica (il nome del cluster):

```
aws elasticache describe-replication-groups --replication-group-id my-cluster
```

Per ulteriori informazioni, consulta l'argomento Per ElastiCache [describe-cache-clusters](#).

Visualizzazione dei dettagli di un cluster (API ElastiCache)

Puoi visualizzare i dettagli di un cluster utilizzando l'operazione DescribeCacheClusters dell'API ElastiCache. Se il parametro CacheClusterId è incluso, vengono restituiti solo i dettagli relativi al cluster specificato. Se si omette il parametro CacheClusterId, vengono restituiti i dettagli relativi a più cluster, fino a un massimo di MaxRecords (valore di default 100). Il valore dei MaxRecords non può essere minore di 20 o maggiore di 100.

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

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheClusters  
&CacheClusterId=my-cluster  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

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

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheClusters  
&MaxRecords=25  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta l'argomento di riferimento dell'API ElastiCache [DescribeCacheClusters](#).

Modifica di un cluster ElastiCache

Oltre all'aggiunta o alla rimozione di nodi da un cluster, può capitare di dover 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 o dopo il riavvio del cluster. Ciò è valido se modifichi il gruppo dei parametri del cluster o un valore di parametro nel gruppo dei parametri del cluster. Per determinare quando viene applicata la modifica di uno specifico parametro, consultare la sezione [Le modifiche diventano effettive della colonna Dettagli nelle tabelle per Parametri specifici di Redis](#).

Utilizzando il AWS Management Console

Per modificare un cluster

1. Accedere AWS Management Console e aprire la ElastiCache console all'[indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Dall'elenco nell'angolo in alto a destra, scegli la AWS regione in cui si trova il cluster che desideri modificare.
3. Nel riquadro di navigazione, scegliere il motore in esecuzione sul cluster da modificare.

Comparirà un elenco dei cluster che eseguono il motore selezionato.

4. Nell'elenco dei cluster, per il cluster che si desidera modificare, scegli il nome.
5. Scegliere Actions (Operazioni), quindi Modify (Modifica).

Comparirà la finestra Modifica cluster.

6. Nella finestra Modify Cluster (Modifica cluster), apportare le modifiche desiderate. Le opzioni includono:
 - Descrizione
 - Modalità cluster: per modificare la modalità cluster da Disabilitata ad Abilitata, è necessario innanzitutto impostare la modalità cluster su Compatibile.

La modalità compatibile consente ai client Redis di connettersi utilizzando la modalità cluster abilitata e la modalità cluster disabilitata. Dopo aver migrato tutti i client Redis per utilizzare

la modalità cluster abilitata, è possibile completare la configurazione della modalità cluster e impostare la modalità cluster su Abilitata.

- Compatibilità versione del motore

Important

Puoi eseguire l'upgrade alle versioni più recenti del motore. Se si aggiornano le versioni principali del motore, ad esempio da 5.0.6 a 6.0, è necessario scegliere una famiglia di gruppi di parametri compatibile con la nuova versione del motore. Per ulteriori informazioni in merito, consulta [Versioni e aggiornamenti del motore](#). Tuttavia, non puoi eseguire il downgrade alle versioni precedenti del motore se non eliminando il cluster esistente e creandolo di nuovo.

- Gruppi di sicurezza VPC
- Gruppo di parametri
- 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](#).

- Multi-AZ
- Failover automatico (solo modalità cluster disabilitata)
- Abilitazione backup automatici.
- Id nodo di backup
- Periodo di retention dei backup
- Finestra di backup
- Argomento per la notifica SNS

La casella Apply Immediately (Applica immediatamente) vale solo per le modifiche delle versioni del motore. Per applicare immediatamente le modifiche, scegliere la casella di controllo Apply Immediately (Applica immediatamente). Se non si seleziona questa casella, le modifiche

del tipo di nodi e delle versioni del motore vengono applicate nella finestra di manutenzione successiva. Le altre modifiche, come quelle della finestra di manutenzione, vengono applicate immediatamente.

7. Scegli Modifica.

Per abilitare/disabilitare la consegna dei registri:

1. Nell'elenco di cluster, scegliere quello da dimensionare. Seleziona Cluster name (Nome del cluster) e non la casella di controllo accanto ad essa.
2. Nella pagina Dettagli cluster, scegli la scheda Registri,
3. Per abilitare/disabilitare i registri lenti, scegli Abilita o Disabilita.

Se scegli l'abilitazione:

- a. Sotto Formato registro, scegli JSON o Testo.
- b. In Tipo di destinazione del registro, scegli CloudWatch Logs o Kinesis Firehose.
- c. In Destinazione del registro, scegli Crea nuovo e inserisci il nome del gruppo di CloudWatchLogs log o il nome dello stream Kinesis Data Firehose. Oppure scegli Seleziona esistente, quindi scegli il nome del gruppo di CloudWatchLogs log o il nome dello stream Kinesis Data Firehose.
- d. Scegli Abilita .

Per modificare la configurazione:

1. Scegliere Modify (Modifica).
2. Sotto Formato registro, scegli JSON o Testo.
3. In Tipo di destinazione, scegli CloudWatch Logs o Kinesis Firehose.
4. In Destinazione del registro, scegli Crea nuovo e inserisci il nome del gruppo di CloudWatchLogs log o il nome dello stream Kinesis Data Firehose. Oppure scegli Seleziona esistente, quindi scegli il nome del gruppo di CloudWatchLogs log o il nome dello stream Kinesis Data Firehose.

Utilizzando il AWS CLI

È possibile modificare un cluster esistente utilizzando l' AWS CLI `modify-cache-cluster` operazione. 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.

Important

Puoi eseguire l'upgrade alle versioni più recenti del motore. Se si aggiornano le versioni principali del motore, ad esempio da 5.0.6 a 6.0, è necessario scegliere una famiglia di gruppi di parametri compatibile con la nuova versione del motore. Per ulteriori informazioni in merito, consulta [Versioni e aggiornamenti del motore](#). Tuttavia, non puoi eseguire il downgrade alle versioni precedenti del motore se non eliminando il cluster esistente e creandolo di nuovo.

Per Linux, macOS o Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --preferred-maintenance-window sun:23:00-mon:02:00
```

Per Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --preferred-maintenance-window sun:23:00-mon:02:00
```

Il parametro `--apply-immediately` si applica solo alle modifiche del tipo di nodi, della versione del motore e del numero dei nodi di un cluster. Se si desidera applicare immediatamente una di queste modifiche, utilizzare il parametro `--apply-immediately`. Se si preferisce posticipare queste modifiche alla prossima finestra di manutenzione, utilizzare il parametro `--no-apply-immediately`. Le altre modifiche, come quelle della finestra di manutenzione, vengono applicate immediatamente.

Per ulteriori informazioni, vedere l' [ElastiCache argomento AWS CLI per `modify-cache-cluster`](#).

Utilizzo dell' ElastiCache API

È possibile modificare un cluster esistente utilizzando l'`ModifyCacheCluster` operazione ElastiCache API. 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.

⚠ Important

Puoi eseguire l'upgrade alle versioni più recenti del motore. Se si aggiornano le versioni principali del motore, ad esempio da 5.0.6 a 6.0, è necessario scegliere una famiglia di gruppi di parametri compatibile con la nuova versione del motore. Per ulteriori informazioni in merito, consulta [Versioni e aggiornamenti del motore](#). Tuttavia, non puoi eseguire il downgrade alle versioni precedenti del motore se non eliminando il cluster esistente e creandolo di nuovo.

Le interruzioni di riga vengono aggiunte per facilitare la lettura.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheCluster  
&CacheClusterId=my-cluster  
&PreferredMaintenanceWindow=sun:23:00-mon:02:00  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150901T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20150202T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20150901T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Il parametro `ApplyImmediately` si applica solo alle modifiche del tipo di nodi, della versione del motore e del numero dei nodi di un cluster. Se si desidera applicare immediatamente una di queste modifiche, impostare il parametro `ApplyImmediately` su `true`. Se si preferisce posticipare queste modifiche alla prossima finestra di manutenzione, impostare il parametro `ApplyImmediately` su `false`. Le altre modifiche, come quelle della finestra di manutenzione, vengono applicate immediatamente.

Per ulteriori informazioni, consulta l'argomento di riferimento sull' [ElastiCache API ModifyCacheCluster](#).

Aggiunta di nodi a un cluster

Per riconfigurare il cluster Redis (modalità cluster abilitata), consulta [Ridimensionamento dei cluster in Redis \(modalità cluster attivata\)](#)

Per aggiungere nodi al cluster si può utilizzare la console di gestione ElastiCache, la AWS CLI o l'API ElastiCache.

Utilizzo di AWS Management Console

Se si vuole aggiungere un nodo a un cluster Redis a nodo singolo (senza funzione di replica abilitata), il processo consiste in due fasi: prima si aggiunge la funzione di replica, poi un nodo di replica.

Argomenti

- [Per aggiungere la funzione di replica a un cluster Redis privo di partizioni](#)
- [Per aggiungere nodi a un cluster \(console\)](#)

La seguente procedura aggiunge la funzione di replica a un cluster Redis a nodo singolo che non ha la funzione di replica abilitata. Quando aggiungi la funzione di replica, il nodo esistente diventa quello primario nel cluster con replica abilitata. Dopo l'aggiunta della funzione di replica, puoi aggiungere al cluster fino a 5 nodi di replica.

Per aggiungere la funzione di replica a un cluster Redis privo di partizioni

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dal riquadro di navigazione scegli Cluster Redis.

Viene visualizzato un elenco dei cluster che eseguono il motore Redis.

3. Scegliere il nome di un cluster, non la casella alla sinistra del nome, a cui aggiungere i nodi.

Quanto segue vale per un cluster Redis che non ha la funzione di replica abilitata:

- Esegue Redis, non Clustered Redis.
- Non presenta partizioni.

Se nel cluster è presente qualche partizione, la funzione di replica è già abilitata e si può proseguire all'indirizzo [Per aggiungere nodi a un cluster \(console\)](#).

4. Scegliere Add replication (Aggiungi funzione di replica).
5. In Add Replication (Aggiungi funzione di replica), digitare una descrizione del cluster con funzione di replica abilitata.
6. Scegli Add (Aggiungi).

Non appena lo stato del cluster ritorna disponibile, puoi passare alla procedura successiva e aggiungere repliche al cluster.

Per aggiungere nodi a un cluster (console)

La procedura seguente può essere utilizzata per aggiungere nodi a un cluster:

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione, scegliere il motore in esecuzione sul cluster a cui aggiungere nodi.

Viene visualizzato un elenco dei cluster che eseguono il motore scelto.

3. Dall'elenco dei cluster, per il cluster a cui si desidera aggiungere un nodo, sceglierne il nome.

Se il cluster è un cluster Redis (modalità cluster abilitata), consulta [Ridimensionamento dei cluster in Redis \(modalità cluster attivata\)](#).

Se il cluster è un cluster Redis (modalità cluster diabilitata), senza partizioni prima completa i passi in [Per aggiungere la funzione di replica a un cluster Redis privo di partizioni](#).

4. Scegliere Add node (Aggiungi nodo).
5. Compilare le informazioni richieste nella finestra di dialogo Add Node (Aggiungi nodo).
6. Scegliere il pulsante Apply Immediately - Yes (Applica immediatamente - Sì) per aggiungere subito il nodo o No per aggiungere il nodo nella prossima finestra di manutenzione del cluster.

Impatto delle nuove richieste di aggiunta e rimozione sulle richieste in sospeso

Scenari	Operazioni in sospeso	Nuova richiesta	Risultati
Scenario 1	Delete	Delete	La nuova richiesta di eliminazione, in sospeso o immediata, sostituisce la richiesta di eliminazione in sospeso.

Scenari	Operazioni in sospeso	Nuova richiesta	Risultati
			<p>Ad esempio, se i nodi 0001, 0003 e 0007 sono in attesa di eliminazione e viene emessa una nuova richiesta di eliminazione dei nodi 0002 e 0004, verranno eliminati solo i nodi 0002 e 0004. I nodi 0001, 0003 e 0007 non verranno eliminati.</p>
Scenario 2	Delete	Crea	<p>La nuova richiesta di creazione, in sospeso o immediata, sostituisce la richiesta di eliminazione in sospeso.</p> <p>Ad esempio, se i nodi 0001, 0003 e 0007 sono in attesa di eliminazione e viene emessa una nuova richiesta di creazione di un nodo, verrà creato un nuovo nodo e i nodi 0001, 0003 e 0007 non verranno eliminati.</p>
Scenario 3	Crea	Delete	<p>La nuova richiesta di eliminazione, in sospeso o immediata, sostituisce la richiesta di creazione in sospeso.</p> <p>Ad esempio, se è presente una richiesta in sospeso per creare due nodi e viene emessa una nuova richiesta per eliminare il nodo 0003, non verranno creati nuovi nodi e il nodo 0003 verrà eliminato.</p>

Scenari	Operazioni in sospenso	Nuova richiesta	Risultati
Scenario 4	Crea	Crea	<p>La nuova richiesta di creazione viene aggiunta alla richiesta di creazione in sospenso.</p> <p>Ad esempio, se è presente una richiesta in sospenso per creare due nodi e viene emessa una nuova richiesta per creare tre nodi, le nuove richieste vengono aggiunte alla richiesta in sospenso e verranno creati cinque nodi.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>⚠ Important</p> <p>Se la nuova richiesta di creazione è impostata su <code>Applica immediatamente</code> - Sì, tutte le richieste di creazione vengono eseguite immediatamente. Se la nuova richiesta di creazione è impostata su <code>Applica immediatamente</code> - No, tutte le richieste di creazione sono in sospenso.</p> </div>

Per determinare quali operazioni sono in sospenso, scegliere l'opzione `Description` (Descrizione) e verificare quante creazioni o eliminazioni in sospenso vengono visualizzate. Non è possibile avere sia creazioni in sospenso che eliminazioni in sospenso.

7. Scegliere il pulsante Add (Aggiungi).

Dopo qualche momento, nell'elenco dovrebbero comparire i nuovi nodi, con lo stato `creating` (in fase di creazione). Se non compaiono, aggiornare la pagina del browser. Quando lo stato del nodo diventa `available`, il nuovo nodo è pronto per l'uso.

Utilizzo di AWS CLI

Se vuoi aggiungere nodi a un cluster esistente che non ha la funzione di replica abilitata, devi prima creare il gruppo di replica specificando il cluster esistente come quello primario. Per ulteriori informazioni, consulta [Creazione di un gruppo di replica utilizzando un cluster di cache Redis](#)

[disponibile \(AWS CLI\)](#) . Una volta che il gruppo di replica è disponibile, si può proseguire con il processo seguente.

Per aggiungere nodi a un cluster tramite l'AWS CLI, utilizzare l'operazione AWS CLI `increase-replica-count` con i seguenti parametri:

- `--replication-group-id` L'ID del gruppo di replica a cui aggiungere nodi.
- `--new-replica-count` specifica il numero di nodi desiderati nel gruppo di replica dopo l'applicazione della modifica. Per aggiungere nodi al cluster, `--new-replica-count` deve essere maggiore del numero attuale di nodi nel cluster.
- `--apply-immediately` o `--no-apply-immediately`, che specificano se aggiungere i nodi subito o alla prossima finestra di manutenzione.

Per Linux, macOS o Unix:

```
aws elasticache increase-replica-count \  
  --replication-group-id my-replication-group \  
  --new-replica-count 4 \  
  --apply-immediately
```

Per Windows:

```
aws elasticache increase-replica-count ^  
  --replication-group-id my-replication-group ^  
  --new-replica-count 4 ^  
  --apply-immediately
```

Questa operazione genera un output simile al seguente (formato JSON):

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "node-test",  
    "Description": "node-test",  
    "Status": "modifying",  
    "PendingModifiedValues": {},  
    "MemberClusters": [  
      "node-test-001",  
      "node-test-002",  
      "node-test-003",  
      "node-test-004",
```

```
    "node-test-005"
  ],
  "NodeGroups": [
    {
      "NodeGroupId": "0001",
      "Status": "modifying",
      "PrimaryEndpoint": {
        "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
        "Port": 6379
      },
      "ReaderEndpoint": {
        "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
        "Port": 6379
      },
      "NodeGroupMembers": [
        {
          "CacheClusterId": "node-test-001",
          "CacheNodeId": "0001",
          "ReadEndpoint": {
            "Address": "node-
test-001.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
          },
          "PreferredAvailabilityZone": "us-west-2a",
          "CurrentRole": "primary"
        },
        {
          "CacheClusterId": "node-test-002",
          "CacheNodeId": "0001",
          "ReadEndpoint": {
            "Address": "node-
test-002.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
          },
          "PreferredAvailabilityZone": "us-west-2c",
          "CurrentRole": "replica"
        },
        {
          "CacheClusterId": "node-test-003",
          "CacheNodeId": "0001",
          "ReadEndpoint": {
            "Address": "node-
test-003.zzzzzz.0001.usw2.cache.amazonaws.com",
            "Port": 6379
          }
        }
      ]
    }
  ]
}
```

```
        },
        "PreferredAvailabilityZone": "us-west-2b",
        "CurrentRole": "replica"
    }
]
}
],
"SnapshottingClusterId": "node-test-002",
"AutomaticFailover": "enabled",
"MultiAZ": "enabled",
"SnapshotRetentionLimit": 1,
"SnapshotWindow": "07:30-08:30",
"ClusterEnabled": false,
"CacheNodeType": "cache.r5.large",
  "DataTiering": "disabled",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false,
"ARN": "arn:aws:elasticache:us-west-2:123456789012:replicationgroup:node-test"
}
}
```

Per ulteriori informazioni, consulta l'argomento della AWS CLI [increase-replica-count](#).

Utilizzo dell'API ElastiCache

Se vuoi aggiungere nodi a un cluster esistente che non ha la funzione di replica abilitata, devi prima creare il gruppo di replica specificando il cluster esistente come quello primario. Per ulteriori informazioni, consulta [Aggiunta di repliche a un cluster Redis standalone \(modalità cluster disabilitata\) \(API ElastiCache\)](#). Una volta che il gruppo di replica è disponibile, si può proseguire con il processo seguente.

Per aggiungere nodi a un cluster (API ElastiCache)

- Chiamare l'operazione API `IncreaseReplicaCount` con i parametri seguenti:
 - `ReplicationGroupId` L'ID del cluster a cui aggiungere nodi.
 - `NewReplicaCount` Il parametro `NewReplicaCount` specifica il numero di nodi nel cluster dopo l'applicazione della modifica. Per aggiungere nodi al cluster, `NewReplicaCount` deve essere maggiore del numero attuale di nodi nel cluster. Se il valore è inferiore al numero attuale di nodi, utilizzare l'API `DecreaseReplicaCount` con un elenco di nodi da rimuovere dal cluster

- **ApplyImmediately** Specifica se aggiungere i nodi subito o alla prossima finestra di manutenzione.
- **Region** Specifica la regione AWS del cluster a cui aggiungere nodi.

L'esempio seguente mostra una chiamata per aggiungere nodi a un cluster.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=IncreaseReplicaCount  
&ApplyImmediately=true  
&NumCacheNodes=4  
&ReplicationGroupId=my-replication-group  
&Region=us-east-2  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Per ulteriori informazioni, consulta l'argomento API ElastiCache [IncreaseReplicaCount](#).

Rimozione di nodi da un cluster

Puoi eliminare un nodo da un cluster utilizzando la AWS Management Console, la AWS CLI o l'API ElastiCache.

Utilizzo di AWS Management Console

Per rimuovere nodi da un cluster (console)

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dall'elenco nell'angolo in alto a destra, scegliere la regione AWS del cluster da cui rimuovere i nodi.
3. Nel pannello di navigazione, scegliere il motore in esecuzione sul cluster da cui rimuovere un nodo.

Viene visualizzato un elenco dei cluster che eseguono il motore scelto.

4. Dall'elenco dei cluster, scegliere il nome del cluster da cui rimuovere un nodo.

Viene visualizzato un elenco dei nodi del cluster.

5. Scegliere la casella a sinistra dell'ID del nodo da rimuovere. Utilizzando la console ElastiCache si può eliminare solo un nodo alla volta, quindi se si selezionano più nodi il pulsante Delete node (Elimina nodo) non può essere utilizzato.

Viene visualizzata la pagina Delete node (Elimina nodo) .

6. Per eliminare il nodo, completare la pagina Delete Node (Elimina nodo) e scegliere Delete Node (Elimina nodo). Per mantenere il nodo, scegliere Cancel (Annulla).

Important

Se l'eliminazione del nodo comporta che il cluster non sia più compatibile con la funzione Multi-AZ, assicurarsi di deselegionare prima la casella di controllo Multi-AZ e quindi eliminare il nodo. Se si deselegiona la casella di controllo Multi-AZ è possibile scegliere di attivare la funzione Auto failover (Failover automatico).

Impatto delle nuove richieste di aggiunta e rimozione sulle richieste in sospeso

Scenari	Operazioni in sospeso	Nuova richiesta	Risultati
Scenario 1	Delete	Delete	<p>La nuova richiesta di eliminazione, in sospeso o immediata, sostituisce la richiesta di eliminazione in sospeso.</p> <p>Ad esempio, se i nodi 0001, 0003 e 0007 sono in attesa di eliminazione e viene emessa una nuova richiesta di eliminazione dei nodi 0002 e 0004, verranno eliminati solo i nodi 0002 e 0004. I nodi 0001, 0003 e 0007 non verranno eliminati.</p>
Scenario 2	Delete	Crea	<p>La nuova richiesta di creazione, in sospeso o immediata, sostituisce la richiesta di eliminazione in sospeso.</p> <p>Ad esempio, se i nodi 0001, 0003 e 0007 sono in attesa di eliminazione e viene emessa una nuova richiesta di creazione di un nodo, verrà creato un nuovo nodo e i nodi 0001, 0003 e 0007 non verranno eliminati.</p>
Scenario 3	Crea	Delete	<p>La nuova richiesta di eliminazione, in sospeso o immediata, sostituisce la richiesta di creazione in sospeso.</p> <p>Ad esempio, se è presente una richiesta in sospeso per creare due nodi e viene emessa una nuova richiesta per eliminare il nodo 0003, non verranno creati nuovi nodi e il nodo 0003 verrà eliminato.</p>
Scenario 4	Crea	Crea	<p>La nuova richiesta di creazione viene aggiunta alla richiesta di creazione in sospeso.</p> <p>Ad esempio, se è presente una richiesta in sospeso per creare due nodi e viene emessa una nuova</p>

Scenari	Operazioni in sospeso	Nuova richiesta	Risultati
			<p>richiesta per creare tre nodi, le nuove richieste vengono aggiunte alla richiesta in sospeso e verranno creati cinque nodi.</p> <div style="border: 1px solid #f08080; padding: 10px; margin-top: 10px;"> <p>⚠ Important</p> <p>Se la nuova richiesta di creazione è impostata su <code>Applica immediatamente - Sì</code>, tutte le richieste di creazione vengono eseguite immediatamente. Se la nuova richiesta di creazione è impostata su <code>Applica immediatamente - No</code>, tutte le richieste di creazione sono in sospeso.</p> </div>

Per determinare quali operazioni sono in sospeso, scegliere l'opzione `Description` (Descrizione) e verificare quante creazioni o eliminazioni in sospeso vengono visualizzate. Non è possibile avere sia creazioni in sospeso che eliminazioni in sospeso.

Utilizzo di AWS CLI

1. Identificare gli ID dei nodi da rimuovere. Per ulteriori informazioni, consulta [Visualizzazione dei dettagli di un cluster](#).
2. Utilizzare l'operazione CLI `decrease-replica-count` 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 `decrease-replica-count` con i seguenti parametri:

- `--replication-group-id` L'ID del gruppo di replica da cui rimuovere i nodi.
- `--new-replica-count` Il parametro `--new-replica-count` specifica il numero di nodi nel cluster dopo l'applicazione della modifica.
- `--replicas-to-remove` Un elenco di ID dei nodi da rimuovere dal cluster.

- `--apply-immediately` o `--no-apply-immediately` Specificano se rimuovere i nodi subito o alla prossima finestra di manutenzione.
- `--region` Specifica la regione AWS del cluster da cui rimuovere i nodi.

Note

È possibile passare solo uno dei parametri `--replicas-to-remove` o `--new-replica-count` quando si chiama questa operazione.

Per Linux, macOS o Unix:

```
aws elasticache decrease-replica-count \  
  --replication-group-id my-replication-group \  
  --new-replica-count 2 \  
  --region us-east-2 \  
  --apply-immediately
```

Per Windows:

```
aws elasticache decrease-replica-count ^  
  --replication-group-id my-replication-group ^  
  --new-replica-count 3 ^  
  --region us-east-2 ^  
  --apply-immediately
```

Questa operazione genera un output simile al seguente (formato JSON):

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "node-test",  
    "Description": "node-test"  
  },  
  "Status": "modifying",  
  "PendingModifiedValues": {},  
  "MemberClusters": [  
    "node-test-001",  
    "node-test-002",  
    "node-test-003",
```

```

        "node-test-004",
        "node-test-005",
        "node-test-006"
    ],
    "NodeGroups": [
        {
            "NodeGroupId": "0001",
            "Status": "modifying",
            "PrimaryEndpoint": {
                "Address": "node-test.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "ReaderEndpoint": {
                "Address": "node-test-
ro.zzzzzz.ng.0001.usw2.cache.amazonaws.com",
                "Port": 6379
            },
            "NodeGroupMembers": [
                {
                    "CacheClusterId": "node-test-001",
                    "CacheNodeId": "0001",
                    "ReadEndpoint": {
                        "Address": "node-
test-001.zzzzzz.0001.usw2.cache.amazonaws.com",
                        "Port": 6379
                    },
                    "PreferredAvailabilityZone": "us-west-2a",
                    "CurrentRole": "primary"
                },
                {
                    "CacheClusterId": "node-test-002",
                    "CacheNodeId": "0001",
                    "ReadEndpoint": {
                        "Address": "node-
test-002.zzzzzz.0001.usw2.cache.amazonaws.com",
                        "Port": 6379
                    },
                    "PreferredAvailabilityZone": "us-west-2c",
                    "CurrentRole": "replica"
                },
                {
                    "CacheClusterId": "node-test-003",
                    "CacheNodeId": "0001",
                    "ReadEndpoint": {

```

```

        "Address": "node-
test-003.zzzzzz.0001.usw2.cache.amazonaws.com",
        "Port": 6379
    },
    "PreferredAvailabilityZone": "us-west-2b",
    "CurrentRole": "replica"
},
{
    "CacheClusterId": "node-test-004",
    "CacheNodeId": "0001",
    "ReadEndpoint": {
        "Address": "node-
test-004.zzzzzz.0001.usw2.cache.amazonaws.com",
        "Port": 6379
    },
    "PreferredAvailabilityZone": "us-west-2c",
    "CurrentRole": "replica"
},
{
    "CacheClusterId": "node-test-005",
    "CacheNodeId": "0001",
    "ReadEndpoint": {
        "Address": "node-
test-005.zzzzzz.0001.usw2.cache.amazonaws.com",
        "Port": 6379
    },
    "PreferredAvailabilityZone": "us-west-2b",
    "CurrentRole": "replica"
},
{
    "CacheClusterId": "node-test-006",
    "CacheNodeId": "0001",
    "ReadEndpoint": {
        "Address": "node-
test-006.zzzzzz.0001.usw2.cache.amazonaws.com",
        "Port": 6379
    },
    "PreferredAvailabilityZone": "us-west-2b",
    "CurrentRole": "replica"
}
    ]
}
],
"SnapshottingClusterId": "node-test-002",

```

```
"AutomaticFailover": "enabled",
"MultiAZ": "enabled",
"SnapshotRetentionLimit": 1,
"SnapshotWindow": "07:30-08:30",
"ClusterEnabled": false,
"CacheNodeType": "cache.r5.large",
  "DataTiering": "disabled",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false,
"ARN": "arn:aws:elasticache:us-west-2:123456789012:replicationgroup:node-
test"
  }
}
```

In alternativa, puoi chiamare `decrease-replica-count` e invece di passare nel parametro `--new-replica-count`, è possibile passare il parametro `--replicas-to-remove`, come illustrato di seguito:

Per Linux, macOS o Unix:

```
aws elasticache decrease-replica-count \
  --replication-group-id my-replication-group \
  --replicas-to-remove node-test-003 \
  --region us-east-2 \
  --apply-immediately
```

Per Windows:

```
aws elasticache decrease-replica-count ^
  --replication-group-id my-replication-group ^
  --replicas-to-remove node-test-003 ^
  --region us-east-2 ^
  --apply-immediately
```

Per ulteriori informazioni, consulta gli argomenti AWS CLI [decrease-replica-count](#).

Utilizzo dell'API ElastiCache

Per rimuovere nodi utilizzando l'API ElastiCache, chiamare l'operazione API

`DecreaseReplicaCount` con l'ID del gruppo di nodi e un elenco dei nodi da rimuovere, nel seguente modo:

- `ReplicationGroupId` L'ID del gruppo di replica da cui rimuovere i nodi.
- `ReplicasToRemove` Il parametro `ReplicasToRemove` specifica il numero di nodi nel cluster dopo l'applicazione della modifica.
- `ApplyImmediately` Specifica se rimuovere i nodi subito o alla prossima finestra di manutenzione.
- `Region` Specifica la regione AWS del cluster da cui rimuovere un nodo.

L'esempio seguente rimuove immediatamente i nodi 0004 e 0005 dal cluster my-cluster.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DecreaseReplicaCount  
  &ReplicationGroupId=my-replication-group  
  &ApplyImmediately=true  
  &ReplicasToRemove=node-test-003  
  &Region us-east-2  
  &Version=2014-12-01  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Per ulteriori informazioni, consulta l'argomento API ElastiCache [DecreaseReplicaCount](#).

Annullamento delle operazioni in attesa di aggiunta e di eliminazione dei nodi

Se hai scelto di non applicare subito una modifica, l'operazione è pending (in attesa) fino a che non viene eseguita nella prossima finestra di manutenzione. Si possono annullare tutte le operazioni in attesa.

Per annullare un'operazione in attesa

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dall'elenco nell'angolo in alto a destra, scegliere la regione AWS in cui si desidera annullare un'operazione in attesa, di aggiunta, o di eliminazione di nodi.
3. Nel pannello di navigazione, scegliere il motore in esecuzione sul cluster in cui sono presenti operazioni in attesa da annullare. Viene visualizzato un elenco dei cluster che eseguono il motore scelto.
4. Nell'elenco dei cluster, scegliere il nome del cluster da modificare, non la casella alla sinistra del nome del cluster, in cui sono presenti operazioni in attesa da annullare.
5. Per determinare quali operazioni sono in sospeso, scegliere l'opzione Description (Descrizione) e verificare quante creazioni o eliminazioni in sospeso vengono visualizzate. Non è possibile avere sia creazioni in sospeso che eliminazioni in sospeso.
6. Scegliere la scheda Nodes (Nodi).
7. Per annullare tutte le operazioni in attesa, fare clic su Cancel Pending (Annulla in attesa). Viene visualizzata la finestra di dialogo Cancel Pending (Annulla in attesa).
8. Confermare la scelta di annullare tutte le operazioni in attesa selezionando il pulsante Cancel Pending (Annulla in attesa); oppure, per mantenere le operazioni, scegliere Cancel (Annulla).

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 elimini un cluster ElastiCache for Redis, vengono conservati i tuoi snapshot manuali. È inoltre possibile creare uno snapshot finale prima che il cluster venga eliminato. Gli snapshot di cache automatici non vengono conservati.

Utilizzo di 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. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di controllo della console ElastiCache, scegliere il motore in esecuzione sul cluster da eliminare.

Viene visualizzato un elenco di tutti i cluster che eseguono quel motore.

3. Per scegliere il cluster da eliminare, seleziona il nome del cluster nell'elenco di cluster.

Important

Puoi eliminare un solo cluster alla volta dalla console ElastiCache. La selezione di più cluster disabilita l'operazione di eliminazione.

4. In Actions (Azioni), scegliere Delete (Elimina).
5. Nella schermata di conferma Delete Cluster (Elimina cluster), scegliere Delete (Elimina) per eliminare il cluster o scegliere Cancel (Annulla) per mantenerlo.

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.

Utilizzo di AWS CLI

Il seguente codice elimina il cluster di cache `my-cluster`.

```
aws elasticache delete-cache-cluster --cache-cluster-id my-cluster
```

L'operazione CLI `delete-cache-cluster` elimina un solo cluster di cache. Per eliminare più cluster di cache, chiamare l'operazione `delete-cache-cluster` per ogni cluster da eliminare. Non occorre attendere la fine dell'eliminazione di un cluster di cache prima di eliminarne un altro.

Per Linux, macOS o Unix:

```
aws elasticache delete-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --region us-east-2
```

Per Windows:

```
aws elasticache delete-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --region us-east-2
```

Per ulteriori informazioni, consulta l'argomento AWS CLI per ElastiCache [delete-cache-cluster](#).

Utilizzo dell'API ElastiCache

Il seguente codice elimina il cluster `my-cluster`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DeleteCacheCluster  
  &CacheClusterId=my-cluster  
  &Region us-east-2  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20150202T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20150202T220302Z
```

```
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

L'operazione API `DeleteCacheCluster` elimina un solo cluster di cache. Per eliminare più cluster di cache, chiamare l'operazione `DeleteCacheCluster` per ogni cluster da eliminare. Non occorre attendere la fine dell'eliminazione di un cluster di cache prima di eliminarne un altro.

Per ulteriori informazioni, consulta l'argomento di riferimento dell'API ElastiCache [DeleteCacheCluster](#).

Accesso al cluster o gruppo di replica

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

Se hai lanciato l'istanza ElastiCache in un Amazon Virtual Private Cloud (Amazon VPC), puoi accedere all'istanza ElastiCache da un'istanza Amazon EC2 nello stesso Amazon VPC. Oppure, utilizzando un peering VPC, puoi accedere all'istanza ElastiCache da un'istanza Amazon EC2 di un Amazon VPC diverso.

Se hai avviato l'istanza ElastiCache in EC2 Classic, consenti all'istanza EC2 di accedere al cluster concedendo al gruppo di sicurezza Amazon EC2 associato al cluster l'accesso al gruppo di sicurezza cache. Per impostazione predefinita, l'accesso a un cluster è riservato all'account che lo ha avviato.

Argomenti

- [Concedi l'accesso al cluster o gruppo di replica](#)

Concedi l'accesso al cluster o gruppo di replica

Il cluster è stato avviato in EC2-VPC.

Se hai lanciato il cluster in un Amazon Virtual Private Cloud (Amazon VPC), puoi connetterti al cluster ElastiCache solo da un'istanza Amazon EC2 in esecuzione nello stesso Amazon VPC. In questo caso sarà necessario concedere al cluster l'ingresso di rete.


Note

Se si utilizza Local Zones, assicurarsi di averla abilitata. Per ulteriori informazioni, consulta [Abilitazione delle zone locali](#). In questo modo, il VPC viene esteso a tale zona locale e il VPC tratterà la sottorete come qualsiasi sottorete in qualsiasi altra Zona di Disponibilità e gateway rilevanti, tabelle di instradamento e altre considerazioni sui gruppi di sicurezza.

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

1. Accedi a AWS Management Console e apri la console Amazon EC2 all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Nel riquadro di navigazione, in Rete e sicurezza, scegli Gruppi di sicurezza.

3. Dall'elenco dei gruppi di sicurezza, scegli il gruppo di sicurezza per il VPC Amazon. A meno che il gruppo di sicurezza non sia stato creato per l'utilizzo di ElastiCache, questo gruppo di sicurezza verrà denominato di default.
4. Scegli la scheda In entrata, quindi procedi come segue:
 - a. Scegli Modifica.
 - b. Scegli Aggiungi regola.
 - c. Nella colonna Tipo, scegli Regola TCP personalizzata.
 - d. Nella casella Intervallo porta, digita 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. Nella casella Fonte, scegli Ovunque che ha l'intervallo di porta (0.0.0.0/0) in modo che qualunque istanza Amazon EC2 avviata in Amazon VPC possa connettersi ai nodi ElastiCache.

 Important

Aprire il cluster ElastiCache 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 ElastiCache. Per ulteriori informazioni, consulta [Gruppi di sicurezza personalizzati](#).

- f. Seleziona Salva.

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

Accesso alle risorse ElastiCache dall'esterno AWS

Amazon ElastiCache è un servizio AWS che fornisce un archivio di tipo chiave-valore in memoria basato sul cloud. Il servizio è progettato per potervi accedere esclusivamente da AWS. Se il cluster ElastiCache è ospitato in un VPC, puoi tuttavia utilizzare un'istanza Network Address Translation (NAT) per fornire accesso dall'esterno.

Requisiti

Per poter accedere alle risorse ElastiCache dall'esterno di AWS è necessario soddisfare i seguenti requisiti:

- Il cluster deve risiedere all'interno di un VPC e deve essere possibile accedervi tramite un'istanza NAT (Network Address Translation). Non ci sono eccezioni a questo requisito.
- L'istanza NAT deve essere avviata nello stesso VPC del cluster.
- L'istanza NAT deve essere avviata nella sottorete pubblica separata dal cluster.
- Un indirizzo IP elastico (EIP) deve essere associato all'istanza NAT. La funzionalità di inoltro della porta iptables viene utilizzata per inoltrare una porta nell'istanza NAT verso la porta del nodo di cache all'interno del VPC.

Considerazioni

Quando si accede alle risorse ElastiCache dall'esterno di ElastiCache, è necessario tenere presenti le seguenti considerazioni.

- I client si connettono all'EIP e alla porta della cache dell'istanza NAT. L'inoltro della porta sull'istanza NAT inoltra il traffico sul nodo cluster di cache appropriato.
- Se un nodo cluster viene aggiunto o sostituito, le regole iptables devono essere aggiornate per riflettere questo cambiamento.

Limitazioni

Questo approccio deve essere utilizzato esclusivamente a scopo di sviluppo e test. Non è consigliato per essere utilizzato per la produzione a causa dei limiti seguenti:

- L'istanza NAT agisce come proxy tra i client e più cluster. L'aggiunta di un proxy ha impatto sulle prestazioni del cluster di cache. L'impatto aumenta con il numero di cluster di cache a cui accedi tramite l'istanza NAT.

- Il traffico dai client verso l'istanza NAT non è crittografato. Devi pertanto evitare l'invio di dati sensibili tramite l'istanza NAT.
- L'istanza NAT aggiunge il sovraccarico della gestione di un'altra istanza.
- L'istanza NAT viene utilizzata come singolo punto di errore. Per informazioni su come configurare l'elevata disponibilità per NAT su VPC, consulta [Elevata disponibilità per istanze NAT Amazon VPC: un esempio](#).

Come accedere alle risorse ElastiCache dall'esterno di AWS

La procedura seguente mostra come eseguire la connessione alle risorse ElastiCache tramite un'istanza NAT.

Questi passaggi presuppongono quanto segue:

- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6380 -j DNAT --to 10.0.1.231:6379`
- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6381 -j DNAT --to 10.0.1.232:6379`

Quindi, è necessario NAT nella direzione opposta:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 10.0.0.55
```

È inoltre necessario abilitare l'inoltro IP, che è disabilitato per impostazione predefinita:

```
sudo sed -i 's/net.ipv4.ip_forward=0/net.ipv4.ip_forward=1/g' /etc/sysctl.conf sudo sysctl --system
```

- Stai accedendo a un cluster Redis con:
 - Indirizzo IP: 10.0.1.230
 - Porta Redis predefinita: 6379
 - Gruppo di sicurezza: sg-bd56b7da
 - Indirizzo IP dell'istanza AWS: sg-bd56b7da
- L'indirizzo IP del client affidabile è 198.51.100.27.
- L'indirizzo IP elastico (EIP) dell'istanza NAT è 203.0.113.73.
- Il gruppo di sicurezza dell'istanza NAT è sg-ce56b7a9.

Per connettersi alle risorse ElastiCache tramite un'istanza NAT

1. Creare un'istanza NAT nello stesso VPC del cluster di cache, ma in una sottorete pubblica.

Per impostazione predefinita, la creazione guidata di VPC avvia un tipo di nodo cache.m1.small. È necessario selezionare una dimensione di nodo in base alle proprie esigenze: È necessario utilizzare l'AMI NAT EC2 per poter accedere a ElastiCache dall'esterno di AWS.

Per informazioni sulla creazione di un'istanza NAT, consulta [Istanze NAT](#) nella Guida per l'utente di AWS VPC.

2. Creare regole del gruppo di sicurezza per il cluster di cache e l'istanza NAT.

Il gruppo di sicurezza per le istanze NAT e l'istanza del cluster devono essere associati alle seguenti regole:

- Due regole in entrata
 - Una per consentire le connessioni TCP dai client affidabili a ciascuna porta della cache inoltrata dall'istanza NAT (6379 - 6381).
 - Un'altra per consentire l'accesso SSH ai client affidabili.

Gruppo di sicurezza per le istanze NAT - Regole in entrata

Type	Protocollo	Intervallo porte	Origine
Regola TCP personalizzata	TCP	6379-6380	198.51.100.27/32
SSH	TCP	22	203.0.113.73/32

- Una regola in uscita per consentire le connessioni TCP alla porta della cache (6379).

Gruppo di sicurezza per le istanze NAT - Regola in uscita

Type	Protocollo	Intervallo porte	Destinazione
Regola TCP personalizzata	TCP	6379	sg-ce56b7a9 (gruppo di sicurezza per le istanze del cluster)

- Una regola in entrata per il gruppo di sicurezza del cluster che consente le connessioni TCP dall'istanza NAT alla porta della cache (6379).

Gruppo di sicurezza per le istanze del cluster - Regola in entrata

Type	Protocollo	Intervallo porte	Origine
Regola TCP personalizzata	TCP	6379	sg-bd56b7da (gruppo di sicurezza del cluster)

3. Convalidare le regole.

- Confermare che il client affidabile possa effettuare la connessione SSH all'istanza NAT.
- Confermare che il client affidabile possa effettuare la connessione al cluster dall'istanza NAT.

4. Aggiungere una regola iptables all'istanza NAT.

È necessario aggiungere una regola iptables alla tabella NAT per ogni nodo nel cluster per inoltrare la porta della cache dall'istanza NAT al nodo del cluster. Di seguito è riportato un possibile esempio:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6379 -j DNAT --to
10.0.1.230:6379
```

Il numero di porta deve essere univoco per ciascun nodo nel cluster. Se ad esempio si utilizza un cluster Redis a tre nodi tramite le porte 6379 - 6381, le regole avrebbero un aspetto simile al seguente:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6379 -j DNAT --to
10.0.1.230:6379
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6380 -j DNAT --to
10.0.1.231:6379
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6381 -j DNAT --to
10.0.1.232:6379
```

5. Confermare che il client affidabile possa effettuare la connessione al cluster.

Il client affidabile deve connettersi all'EIP associato all'istanza NAT e alla porta del cluster corrispondente al nodo del cluster appropriato. Ad esempio, la stringa di connessione per PHP potrebbe apparire come segue:

```
redis->connect( '203.0.113.73', 6379 );  
redis->connect( '203.0.113.73', 6380 );  
redis->connect( '203.0.113.73', 6381 );
```

È anche possibile utilizzare un client telnet per verificare la connessione. Per esempio:

```
telnet 203.0.113.73 6379  
telnet 203.0.113.73 6380  
telnet 203.0.113.73 6381
```

6. Salvare la configurazione iptables.

Salvare le regole dopo averle sottoposte a test e a verifica. Se si sta utilizzando una distribuzione Linux basata su Redhat (ad esempio Amazon Linux), eseguire questo comando:

```
service iptables save
```

Argomenti correlati

Negli argomenti seguenti vengono fornite informazioni aggiuntive.

- [Modelli di accesso a una cache ElastiCache in un Amazon VPC](#)
- [Accesso a una cache ElastiCache da un'applicazione in esecuzione nel data center di un cliente](#)
- [Istanze NAT](#)
- [Configurazione di client ElastiCache](#)
- [Elevata disponibilità per istanze NAT Amazon VPC: un esempio](#)

Individuazione degli endpoint di connessione

Le applicazioni utilizzano gli endpoint per connettersi ai cluster. Un endpoint corrisponde all'indirizzo univoco di un nodo o di un cluster.

In caso contrario, occorre configurare il client in modo che si avvalga degli endpoint di nodo individuali per le operazioni di lettura e scrittura. È necessario, inoltre, tenere traccia dei nodi eventualmente aggiunti o rimossi.

Quali endpoint utilizzare

- Nodo autonomo di Redis; prevede l'utilizzo dell'endpoint del nodo per entrambe le operazioni di lettura e scrittura.
- Cluster Redis (modalità cluster disabilitata); prevedono l'utilizzo dell'endpoint primario per tutte le operazioni di scrittura. Usa l'endpoint di lettura per dividere equamente le connessioni in arrivo verso l'endpoint tra tutte le repliche di lettura. I vari endpoint di nodo (definiti in ambiente API/CLI endpoint di lettura) vengono invece impiegati per le operazioni di lettura.
- Cluster Redis (modalità cluster abilitata); prevedono l'utilizzo dell'endpoint di configurazione del cluster per tutte le operazioni che supportano i comandi abilitati in modalità cluster. È necessario utilizzare un client che supporti il cluster Redis (Redis 3.2). È comunque possibile effettuare letture dagli endpoint di nodo individuali (definiti in ambiente API/CLI endpoint di lettura).

Nelle sezioni seguenti è illustrato, fase per fase, il processo di individuazione degli endpoint necessari per il motore in esecuzione.

Individuazione degli endpoint di un cluster Redis (modalità cluster disabilitata) (console)

Se un cluster Redis (modalità cluster disabilitata) dispone di un solo nodo, l'endpoint di quest'ultimo viene utilizzato per entrambe le operazioni di lettura e scrittura. Se un cluster Redis (modalità cluster disabilitata) dispone di più nodi, ci sono tre tipi di endpoint, l'endpoint primario, l'endpoint di lettura e gli endpoint dei nodi.

L'endpoint primario è un nome DNS che viene sempre risolto nel nodo primario nel cluster. L'endpoint primario è immune a modifiche al cluster, ad esempio la promozione di una replica di lettura al ruolo primario. Per attività di scrittura, è preferibile collegare le applicazioni all'endpoint primario.

Un endpoint di lettura dividerà equamente le connessioni in arrivo sull'endpoint tra tutte le repliche di lettura in un cluster ElastiCache for Redis. Ulteriori fattori come, ad esempio, quando l'applicazione crea le connessioni o il modo in cui l'applicazione riutilizza le connessioni determineranno la distribuzione del traffico. Gli endpoint di lettura mantengono le modifiche ai cluster in tempo reale mano a mano che le repliche vengono aggiunte o rimosse. Puoi posizionare le repliche di lettura multiple del cluster ElastiCache for Redis in diverse zone di disponibilità (AZ) AWS per garantire un'elevata disponibilità degli endpoint di lettura.

Note

Un endpoint di lettura non è un servizio di load balancer. Si tratta di un record DNS che si risolverà in un indirizzo IP di uno dei nodi di replica in modo round robin.

Per attività di lettura, le applicazioni possono inoltre connettersi a qualsiasi nodo nel cluster. A differenza dell'endpoint primario, gli endpoint dei nodi vengono risolti in endpoint specifici. Se apporti una modifica al cluster, ad esempio aggiungendo o eliminando una replica, devi aggiornare gli endpoint dei nodi nell'applicazione.

Per individuare gli endpoint di un cluster Redis (modalità cluster disabilitata)

1. Accedi alla AWS Management Console e apri la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dal riquadro di navigazione scegli Cluster Redis.

Verrà visualizzata la schermata con l'elenco dei cluster Redis (modalità cluster disabilitata) e Redis (modalità cluster abilitata).

3. Per trovare l'endpoint primario e/o l'endpoint di lettura del cluster, scegli il nome del cluster (non il pulsante alla sua sinistra).

▼ Cluster details

Cluster name [redacted]	Description [redacted]	Node type cache.r6g.large	Status Available
Engine Redis	Engine version 6.0.5	Global datastore -	Global datastore role -
Update status Update available	Cluster mode Off	Shards 1	Number of nodes 3
Data tiering Disabled	Multi-AZ Enabled	Auto-failover Enabled	Encryption in transit Disabled
Encryption at rest Disabled	Parameter group default.redis6.x	Outpost ARN -	Configuration endpoint -
Primary endpoint [redacted]-encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	Reader endpoint [redacted]-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com:6379	ARN [redacted]	

Endpoint primario e di lettura per un cluster Redis (modalità cluster disabilitata)

Se il cluster annovera un solo nodo, non c'è alcun endpoint primario e si può passare alla fase successiva.

4. Se il cluster Redis (modalità cluster disabilitata) dispone di nodi di replica, puoi trovare gli endpoint del nodo di replica scegliendo il nome del cluster e quindi selezionando la scheda Nodes (Nodi).

Così facendo, viene visualizzata una schermata con l'elenco di tutti i nodi del cluster, primario e di replica, e i relativi endpoint.

<input type="checkbox"/>	Node Name	Status	Current Role	Port	Endpoint
<input type="checkbox"/>	test-no-001	available	primary	6379	[redacted]amazonaws.com
<input type="checkbox"/>	test-no-002	available	replica	6379	[redacted]amazonaws.com
<input type="checkbox"/>	test-no-003	available	replica	6379	[redacted]amazonaws.com

Endpoint di nodo di un cluster Redis (modalità cluster disabilitata)

5. Per copiare un endpoint negli appunti:
 - a. Individua, uno alla volta, gli endpoint da copiare.
 - b. Scegli l'icona di copia direttamente davanti all'endpoint.

L'endpoint è stato correttamente copiato negli appunti. Per informazioni sull'utilizzo dell'endpoint per la connessione a un nodo, vedere [Connessione ai nodi](#).

Un endpoint primario di Redis (modalità cluster disattivata) risulta simile a quello riportato di seguito. Differisce in base all'avvenuta o mancata abilitazione della crittografia dati in transito.

Crittografia dati in transito non abilitata

```
clusterName.xxxxxx.nodeId.regionAndAz.cache.amazonaws.com:port
```

```
redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
```

Crittografia dati in transito abilitata

```
master.clusterName.xxxxxx.regionAndAz.cache.amazonaws.com:port
```

```
master.ncit.ameaqx.use1.cache.amazonaws.com:6379
```

Individuazione degli endpoint di un cluster Redis (modalità cluster abilitata) (console)

Un cluster Redis (modalità cluster abilitata) dispone di un solo endpoint di configurazione. Connettendosi all'endpoint di configurazione, la tua applicazione è in grado di scoprire gli endpoint primari e di lettura per ogni partizione del cluster.

Per trovare l'endpoint di un cluster Redis (modalità cluster abilitata)

1. Accedi alla AWS Management Console e apri la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dal riquadro di navigazione scegli Cluster Redis.

Verrà visualizzata la schermata con l'elenco dei cluster Redis (modalità cluster disabilitata) e Redis (modalità cluster abilitata). Scegli il cluster Redis (modalità cluster abilitata) a cui effettuare il collegamento.

3. Per trovare l'endpoint di configurazione del cluster, scegli il nome del cluster (non il pulsante di opzione).
4. L'endpoint di configurazione è visualizzato sotto i dettagli del cluster. Per copiarlo scegli l'icona copia a sinistra dell'endpoint.

Individuazione degli endpoint (AWS CLI)

Per trovare gli endpoint di nodi, cluster e gruppi di replica, è possibile avvalersi di AWS CLI per Amazon ElastiCache.

Argomenti

- [Individuazione degli endpoint per nodi e cluster \(AWS CLI\)](#)
- [Individuazione degli endpoint per gruppi di replica \(AWS CLI\)](#)

Individuazione degli endpoint per nodi e cluster (AWS CLI)

AWS CLI, con il comando `describe-cache-clusters`, consente di individuare gli endpoint di un cluster con i relativi nodi. Per i cluster Redis, il comando restituisce l'endpoint del cluster. Se si include il parametro facoltativo `--show-cache-node-info`, il comando restituisce, inoltre, gli endpoint dei vari nodi nel cluster.

Example

Il seguente comando recupera le informazioni relative al cluster a nodo singolo `mycluster`.

Important

Il parametro `--cache-cluster-id` può essere utilizzato con l'ID del cluster a nodo singolo o con gli ID dei nodi nei gruppi di replica Redis. Il parametro `--cache-cluster-id` di un gruppo di replica Redis è rappresentato da un valore di quattro cifre, ad esempio `0001`. Se `--cache-cluster-id` rappresenta l'ID di un cluster (nodo) in un gruppo di replica Redis, il parametro `replication-group-id` è incluso nell'output.

Per Linux, macOS o Unix:

```
aws elasticache describe-cache-clusters \  
  --cache-cluster-id redis-cluster \  
  --show-cache-node-info
```

Per Windows:

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id redis-cluster ^
```

```
--show-cache-node-info
```

L'output dell'operazione sopra indicata dovrebbe essere simile a questo (in formato JSON).

```
{
  "CacheClusters": [
    {
      "CacheClusterStatus": "available",
      "SecurityGroups": [
        {
          "SecurityGroupId": "sg-77186e0d",
          "Status": "active"
        }
      ],
      "CacheNodes": [
        {
          "CustomerAvailabilityZone": "us-east-1b",
          "CacheNodeCreateTime": "2018-04-25T18:19:28.241Z",
          "CacheNodeStatus": "available",
          "CacheNodeId": "0001",
          "Endpoint": {
            "Address": "redis-cluster.amazonaws.com",
            "Port": 6379
          },
          "ParameterGroupStatus": "in-sync"
        }
      ],
      "AtRestEncryptionEnabled": false,
      "CacheClusterId": "redis-cluster",
      "TransitEncryptionEnabled": false,
      "CacheParameterGroup": {
        "ParameterApplyStatus": "in-sync",
        "CacheNodeIdsToReboot": [],
        "CacheParameterGroupName": "default.redis3.2"
      },
      "NumCacheNodes": 1,
      "PreferredAvailabilityZone": "us-east-1b",
      "AutoMinorVersionUpgrade": true,
      "Engine": "redis",
      "AuthTokenEnabled": false,
      "PendingModifiedValues": {},
      "PreferredMaintenanceWindow": "tue:08:30-tue:09:30",
      "CacheSecurityGroups": [],
    }
  ]
}
```

```
        "CacheSubnetGroupName": "default",
        "CacheNodeType": "cache.t2.small",
        "DataTiering": "disabled"
        "EngineVersion": "3.2.10",
        "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
        "CacheClusterCreateTime": "2018-04-25T18:19:28.241Z"
    }
]
}
```

Per ulteriori informazioni, consulta l'argomento [describe-cache-clusters](#).

Individuazione degli endpoint per gruppi di replica (AWS CLI)

AWS CLI, con il comando `describe-replication-groups`, consente di individuare gli endpoint di un gruppo di replica e i relativi cluster. Il comando restituisce l'endpoint primario del gruppo di replica e l'elenco di tutti i cluster (nodi) nel gruppo di replica con i loro endpoint insieme all'endpoint di lettura.

L'operazione seguente recupera l'endpoint primario e l'endpoint di lettura per il gruppo di replica `myreplgroup`. Utilizza l'endpoint primario per tutte le operazioni di scrittura.

```
aws elasticache describe-replication-groups \
  --replication-group-id myreplgroup
```

Per Windows:

```
aws elasticache describe-replication-groups ^
  --replication-group-id myreplgroup
```

L'aspetto dell'output di questa operazione è simile al seguente (in formato JSON).

```
{
  "ReplicationGroups": [
    {
      "Status": "available",
      "Description": "test",
      "NodeGroups": [
        {
          "Status": "available",
          "NodeGroupMembers": [
```

```
    {
      "CurrentRole": "primary",
      "PreferredAvailabilityZone": "us-west-2a",
      "CacheNodeId": "0001",
      "ReadEndpoint": {
        "Port": 6379,
        "Address": "myreplgroup-001.amazonaws.com"
      },
      "CacheClusterId": "myreplgroup-001"
    },
    {
      "CurrentRole": "replica",
      "PreferredAvailabilityZone": "us-west-2b",
      "CacheNodeId": "0001",
      "ReadEndpoint": {
        "Port": 6379,
        "Address": "myreplgroup-002.amazonaws.com"
      },
      "CacheClusterId": "myreplgroup-002"
    },
    {
      "CurrentRole": "replica",
      "PreferredAvailabilityZone": "us-west-2c",
      "CacheNodeId": "0001",
      "ReadEndpoint": {
        "Port": 6379,
        "Address": "myreplgroup-003.amazonaws.com"
      },
      "CacheClusterId": "myreplgroup-003"
    }
  ],
  "NodeGroupId": "0001",
  "PrimaryEndpoint": {
    "Port": 6379,
    "Address": "myreplgroup.amazonaws.com"
  },
  "ReaderEndpoint": {
    "Port": 6379,
    "Address": "myreplgroup-ro.amazonaws.com"
  }
}
],
"ReplicationGroupId": "myreplgroup",
"AutomaticFailover": "enabled",
```

```
    "SnapshottingClusterId": "myreplgroup-002",
    "MemberClusters": [
      "myreplgroup-001",
      "myreplgroup-002",
      "myreplgroup-003"
    ],
    "PendingModifiedValues": {}
  }
]
}
```

Per ulteriori informazioni, consulta [describe-replication-groups](#) nella documentazione di riferimento dell'AWS CLI.

Ricerca degli endpoint (API ElastiCache)

Per trovare gli endpoint di nodi, cluster e gruppi di replica, è possibile avvalersi dell'API di Amazon ElastiCache.

Argomenti

- [Individuazione degli endpoint di nodi e cluster \(API ElastiCache\)](#)
- [Individuazione di endpoint dei gruppi di replica \(API ElastiCache\)](#)

Individuazione degli endpoint di nodi e cluster (API ElastiCache)

L'API di ElastiCache, con l'operazione `DescribeCacheClusters`, consente di individuare gli endpoint di un cluster con i relativi nodi. Per i cluster Redis, il comando restituisce l'endpoint del cluster. Se si include il parametro facoltativo `ShowCacheNodeInfo`, l'operazione restituisce, inoltre, gli endpoint dei vari nodi nel cluster.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheClusters  
  &CacheClusterId=mycluster  
  &ShowCacheNodeInfo=true  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &Version=2015-02-02  
  &X-Amz-Credential=<credential>
```

Individuazione di endpoint dei gruppi di replica (API ElastiCache)

L'API di ElastiCache, con l'operazione `DescribeReplicationGroups`, consente di individuare gli endpoint di un gruppo di replica e i relativi cluster. Il comando restituisce l'endpoint primario del gruppo di replica e l'elenco di tutti i cluster nel gruppo di replica con i loro endpoint insieme all'endpoint di lettura.

La seguente operazione recupera l'endpoint primario (`PrimaryEndpoint`), l'endpoint di lettura (`ReaderEndpoint`) e gli endpoint dei singoli nodi (`ReadEndpoint`) relativi al gruppo di replica `myrep1group`. Utilizza l'endpoint primario per tutte le operazioni di scrittura.

```
https://elasticache.us-west-2.amazonaws.com/
```

```
?Action=DescribeReplicationGroups
&ReplicationGroupId=myreplgroup
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta [DescribeReplicationGroups](#)

Utilizzo degli shard.

Uno shard (API/CLI: gruppo di nodi) è una raccolta composta da uno a nodi Redis. Un cluster Redis (modalità cluster disabilitata) non disporrà mai di più shard. Con gli shard, puoi separare i database di grandi dimensioni in parti più piccole, più veloci e più facili da gestire chiamate frammenti di dati. Ciò può aumentare l'efficienza del database distribuendo le operazioni su più sezioni separate. L'utilizzo degli shard può offrire molti vantaggi, tra cui miglioramento delle prestazioni, della scalabilità e dell'efficienza dei costi.

Puoi creare un cluster con un numero più alto di partizioni e un numero più basso di repliche per un totale di 90 nodi per cluster. Questa configurazione del cluster può andare da 90 partizioni e 0 repliche a 15 partizioni e 5 repliche che è il numero massimo consentito di repliche. I dati del cluster vengono partizionati tra gli shard del cluster. Se uno shard contiene più nodi, lo shard implementa la replica con un nodo che agisce da nodo primario lettura/scrittura e gli altri nodi da nodi di replica di sola lettura.

Il limite del nodo o della partizione può essere aumentato fino a un massimo di 500 per cluster se la versione del motore Redis è 5.0.6 o superiore. Ad esempio, è possibile scegliere di configurare un cluster a 500 nodi che varia tra 83 partizioni (un primario e 5 repliche per partizione) e 500 partizioni (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. Per ulteriori informazioni, consulta [Creazione di un gruppo di sottoreti](#).

Per le versioni inferiori alla 5.0.6, il limite è 250 per cluster.

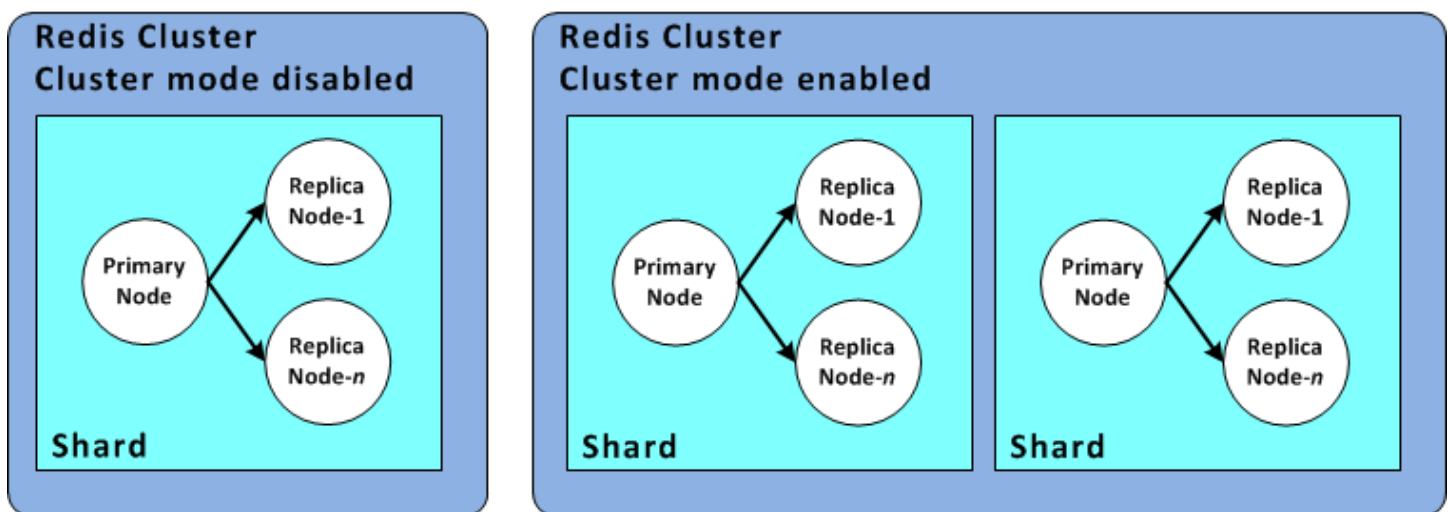
Per richiedere un aumento dei limiti, consulta [AWS Limiti dei servizi AWS](#) e seleziona il tipo di limite Nodi per cluster per tipo di istanza.

Quando si crea un cluster Redis (modalità cluster abilitata) utilizzando la ElastiCache console, si specifica il numero di shard nel cluster e il numero di nodi negli shard. Per ulteriori informazioni,

consulta [Creazione di un cluster Redis \(modalità cluster abilitata\) \(console\)](#). Se utilizzi l' ElastiCache API o AWS CLI per creare un cluster (chiamato gruppo di replica nell'API/CLI), puoi configurare il numero di nodi in uno shard (API/CLI: gruppo di nodi) in modo indipendente. Per ulteriori informazioni, consulta gli argomenti seguenti:

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

Ogni nodo in uno shard presenta le stesse specifiche di calcolo, storage e memoria. L' ElastiCache API consente di controllare gli attributi a livello di shard, come il numero di nodi, le impostazioni di sicurezza e le finestre di manutenzione del sistema.



Configurazioni shard Redis

Per ulteriori informazioni, consultare [Reshard offline e ribilanciamento dello shard per Redis \(modalità cluster abilitata\)](#) e [Reshard online e ribilanciamento della partizione per Redis \(modalità cluster abilitata\)](#).

Ricerca dell'ID di uno shard

Puoi trovare l'ID di uno shard utilizzando AWS Management Console, the AWS CLI o l'API. ElastiCache

Utilizzando il AWS Management Console

Argomenti

- [Redis \(modalità cluster disattivata\)](#)

- [Redis \(Modalità cluster abilitata\)](#)

Redis (modalità cluster disattivata)

Per i gruppi di replica Redis (modalità cluster disabilitata), l'ID dello shard è sempre 0001.

Redis (Modalità cluster abilitata)

La procedura seguente utilizza AWS Management Console per trovare l'ID shard del gruppo di replica di Redis (modalità cluster abilitata).

Per individuare l'ID shard in un gruppo di replica Redis (modalità cluster abilitata)

1. [Accedi AWS Management Console e apri la ElastiCache console all'indirizzo https://console.aws.amazon.com/elasticache/.](https://console.aws.amazon.com/elasticache/)
2. Dal pannello di navigazione, scegliere Redis, quindi scegliere il nome del gruppo di replica Redis (modalità cluster abilitata) per il quale si desidera trovare l'ID shard.
3. Nella colonna Shard Name (Nome shard), l'ID shard sono le ultime quattro cifre del nome dello shard.

Utilizzando il AWS CLI

Per trovare gli ID shard (gruppo di nodi) per i gruppi di replica Redis (modalità cluster disabilitata) o Redis (modalità cluster abilitata), utilizzate l' AWS CLI operazione `describe-replication-groups` con il seguente parametro opzionale.

- **--replication-group-id**- Parametro facoltativo che, se utilizzato, limita l'output ai dettagli del gruppo di replica specificato. Se si omette il parametro, vengono restituiti i dettagli di un massimo di 100 gruppi di replica.

Example

Questo comando restituisce i dettagli per `sample-repl-group`.

Per Linux, macOS o Unix:

```
aws elasticache describe-replication-groups \  
  --replication-group-id sample-repl-group
```

Per Windows:

```
aws elasticache describe-replication-groups ^  
  --replication-group-id sample-repl-group
```

L'aspetto dell'output di questo comando è simile al seguente. Gli ID shard (gruppo di nodi) sono *evidenziati* qui per semplificare la ricerca.

```
{  
  "ReplicationGroups": [  
    {  
      "Status": "available",  
      "Description": "2 shards, 2 nodes (1 + 1 replica)",  
      "NodeGroups": [  
        {  
          "Status": "available",  
          "Slots": "0-8191",  
          "NodeGroupId": "0001",  
          "NodeGroupMembers": [  
            {  
              "PreferredAvailabilityZone": "us-west-2c",  
              "CacheNodeId": "0001",  
              "CacheClusterId": "sample-repl-group-0001-001"  
            },  
            {  
              "PreferredAvailabilityZone": "us-west-2a",  
              "CacheNodeId": "0001",  
              "CacheClusterId": "sample-repl-group-0001-002"  
            }  
          ]  
        },  
        {  
          "Status": "available",  
          "Slots": "8192-16383",  
          "NodeGroupId": "0002",  
          "NodeGroupMembers": [  
            {  
              "PreferredAvailabilityZone": "us-west-2b",  
              "CacheNodeId": "0001",  
              "CacheClusterId": "sample-repl-group-0002-001"  
            },  
            {  
              "PreferredAvailabilityZone": "us-west-2a",
```

```

        "CacheNodeId": "0001",
        "CacheClusterId": "sample-repl-group-0002-002"
    }
  ]
},
"ConfigurationEndpoint": {
  "Port": 6379,
  "Address": "sample-repl-
group.9dcv5r.clustercfg.usw2.cache.amazonaws.com"
},
"ClusterEnabled": true,
"ReplicationGroupId": "sample-repl-group",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "enabled",
"SnapshotWindow": "13:00-14:00",
"MemberClusters": [
  "sample-repl-group-0001-001",
  "sample-repl-group-0001-002",
  "sample-repl-group-0002-001",
  "sample-repl-group-0002-002"
],
"CacheNodeType": "cache.m3.medium",
"DataTiering": "disabled",
"PendingModifiedValues": {}
}
]
}

```

Utilizzo dell'API ElastiCache

Per trovare gli ID shard (gruppo di nodi) per i gruppi di replica Redis (modalità cluster disabilitata) o Redis (modalità cluster abilitata), utilizzate l' AWS CLI operazione `describe-replication-groups` con il seguente parametro opzionale.

- **ReplicationGroupId**- Parametro facoltativo che, se utilizzato, limita l'output ai dettagli del gruppo di replica specificato. Se si omette il parametro, vengono restituiti i dettagli di un massimo di `xxx` gruppi di replica.

Example

Questo comando restituisce i dettagli per `sample-repl-group`.

Per Linux, macOS o Unix:

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroup  
&ReplicationGroupId=sample-repl-group  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Confronto tra cache progettate autonomamente Memcached e Redis

Amazon ElastiCache supporta i motori cache Memcached e Redis. Ogni motore offre alcuni vantaggi. Utilizza le informazioni in questo argomento per individuare il motore e la versione che meglio si adattano alle tue esigenze.

Important

Dopo aver creato un cluster di cache o un gruppo di replica, puoi aggiornare a una versione del motore più recente, ma non puoi eseguire il downgrade a versioni motore precedenti. Se vuoi utilizzare una versione del motore precedente, elimina il cluster di cache o il gruppo di replica esistente e crealo di nuovo con la versione del motore precedente.

A prima vista, i motori sembrano simili. Ognuno di essi è uno store chiave-valore in memoria. Tuttavia, vi sono alcune differenze importanti.

Scegliere Memcached per le seguenti esigenze:

- Hai bisogno del modello in assoluto più semplice.
- Hai necessità di eseguire grandi nodi con più core o thread.
- Hai la possibilità di dimensionare, aggiungere e rimuovere i nodi a seconda che la domanda del tuo sistema sia crescente o decrescente.
- Hai necessità di memorizzare gli oggetti nella cache.

Scegliere Redis con una versione di ElastiCache se si applicano le seguenti condizioni:

- ElastiCache per Redis versione 7.0 (avanzata)

Si desidera utilizzare [Funzioni Redis](#), [Pub/Sub partizionato](#) o [Miglioramenti ACL Redis](#). Per ulteriori informazioni, consultare [Redis Versione 7.0 \(avanzata\)](#).

- ElastiCache for Redis versione 6.2 (avanzata)

Vuoi la possibilità di eseguire il livello dei dati tra memoria e SSD utilizzando il tipo di nodo r6gd. Per ulteriori informazioni, consulta la sezione [Tiering di dati](#).

- ElastiCache for Redis versione 6.0 (avanzata)

Vuoi autenticare gli utenti con il controllo accessi basato sui ruoli.

Per ulteriori informazioni, consulta la sezione [Redis Versione 6.0 \(avanzata\)](#).

- ElastiCache for Redis versione 5.0.0 (potenziata)

Vuoi utilizzare i [flussi Redis](#), una struttura di registro dei dati che consente ai produttori di accodare nuove voci in tempo reale e che consente ai consumatori di utilizzare messaggi con blocchi o senza blocchi.

Per ulteriori informazioni, consulta la sezione [Redis versione 5.0.0 \(potenziata\)](#).

- ElastiCache for Redis versione 4.0.10 (potenziata)


Supporta la crittografia e l'aggiunta o la rimozione automatica degli shard dal cluster Redis (modalità cluster abilitata).

Per ulteriori informazioni, consulta la sezione [Redis versione 4.0.10 \(potenziata\)](#).

Le seguenti versioni sono obsolete, hanno raggiunto o stanno per giungere a fine vita.

- ElastiCache for Redis versione 3.2.10 (potenziata)

Supporta la capacità di aggiungere o rimuovere dinamicamente gli shard dal cluster Redis (modalità cluster abilitata).

 Important

Al momento ElastiCache for Redis 3.2.10 non supporta la crittografia.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Redis versione 3.2.10 \(potenziata\)](#)
- Best practice per il resharding online di Redis, per ulteriori informazioni consultare:
 - [Best practice: resharding online](#)
 - [Resharding online e ribilanciamento partizioni per Redis \(modalità cluster abilitata\)](#)
- Per ulteriori informazioni sul dimensionamento dei cluster Redis, consulta [Dimensionamento](#).
- ElastiCache for Redis versione 3.2.6 (potenziata)

Se sono necessarie la caratteristiche delle versioni Redis precedenti e le caratteristiche seguenti, scegliere ElastiCache for Redis 3.2.6:

- Crittografia dei dati in transito Per ulteriori informazioni consulta [Crittografia dei dati in transito di Amazon ElastiCache for Redis](#).
- Crittografia dei dati inattivi Per ulteriori informazioni, consulta [Crittografia dei dati inattivi di Amazon ElastiCache per Redis](#).
- ElastiCache for Redis (modalità cluster abilitata) versione 3.2.4

Se sono necessarie la caratteristiche delle versioni Redis 2.8.x e le caratteristiche seguenti, scegliere Redis 3.2.4 (modalità cluster):

- Hai la necessità di eseguire la partizione dei dati tra due fino a 500 gruppi di nodi (solo in modalità cluster).
- Hai la necessità di effettuare l'indicizzazione geospaziale (modalità cluster o non-cluster).
- Non è necessario il support di più database.
- ElastiCache for Redis (modalità non cluster) 2.8.x e 3.2.4 (potenziata)

Scegliere Redis 2.8.x o Redis 3.2.4 (modalità non cluster) se si applicano le seguenti condizioni:

- Sono necessari tipi di dati complessi, ad esempio stringa, set, set ordinati, elenchi, hash e bitmap.
- È necessario ordinare o classificare i dataset in-memory.
- Il tuo store di chiavi deve essere permanente.
- È necessario replicare i dati dai primari a quelli relativi a una o più repliche di lettura per applicazioni di letture intensive.
- Se il primo nodo riporta un errore, è necessario un failover automatico.

- È necessario pubblicare e sottoscrivere le caratteristiche (pub/sub) per informare i clienti sugli eventi sul server.
- Sono necessarie le caratteristiche di backup e ripristino.
- È necessario il supporto di più database.

Riepilogo del confronto di Memcached, Redis (modalità cluster disabilitata) e Redis (modalità cluster abilitata)

	Memcached	Redis (modalità cluster disabilitata)	Redis (modalità cluster abilitata)
Engine versions+	1.4.5 and later	4.0.10 and later	4.0.10 and later
Data types	Simple	2.8.x - Complex * Complex	3.2.x and later - Complex
Data partitioning	Yes	No	Yes
Cluster is modifiable	Yes	Yes	3.2.10 and later - Limited
Online resharding	No	No	3.2.10 and later
Encryption	in-transit 1.6.12 and later	4.0.10 and later	4.0.10 and later
Data tiering	No	6.2 and later	6.2 and later
Certificazioni di conformità			
Compliance Certification			
FedRAMP	Sì - 1.6.12 e versioni successive	versioni 4.0.10 e successive	versioni 4.0.10 e successive
HIPAA	Sì - 1.6.12 e versioni successive	versioni 4.0.10 e successive	versioni 4.0.10 e successive
PCI DSS	Sì	versioni 4.0.10 e successive	versioni 4.0.10 e successive
Multi-threaded	Yes	No	No
Node type upgrade	No	Yes	Yes
Engine upgrading	Yes	Yes	Yes

	Memcached	Redis (modalità cluster disabilitata)	Redis (modalità cluster abilitata)
High availability (replication)	No	Yes	Yes
Automatic failover	No	Optional	Required
Pub/Sub capabilities	No	Yes	Yes
Sorted sets	No	Yes	Yes
Backup and restore	No	Yes	Yes
Geospatial indexing	No	4.0.10 and later	Yes

Note:

string, objects (like databases)

* string, sets, sorted sets, lists, hashes, bitmaps, hyperloglog

string, sets, sorted sets, lists, hashes, bitmaps, hyperloglog, geospatial indexes

+ Excludes versions which are deprecated, have reached or soon to reach end of life.

Dopo aver scelto il motore per il cluster, ti consigliamo di utilizzare la versione più recente per quel motore. Per ulteriori informazioni, consulta [Versioni supportate di ElastiCache for Memcached](#) o [Versioni supportate di ElastiCache for Redis](#).

Migrazione online a ElastiCache

Utilizzando la migrazione online, è possibile migrare i dati da Redis open source in self-hosting su Amazon EC2 ad Amazon ElastiCache.

Note

La migrazione online non è supportata per la cache serverless o i cluster ElastiCache in esecuzione sul tipo di nodo r6gd.

Panoramica

Per migrare i dati da Redis open source in esecuzione su Amazon EC2 ad Amazon ElastiCache, è necessaria un'implementazione Amazon ElastiCache esistente o appena creata. Questa distribuzione deve avere una configurazione pronta per la migrazione. Deve anche essere in linea con la configurazione desiderata, inclusi gli attributi quali il tipo di istanza, il numero di partizioni e il numero di repliche.

La migrazione online è progettata per migrare i dati da Redis open source in self-hosting su Amazon EC2 a ElastiCache per Redis e non tra cluster ElastiCache per Redis.

Important

Si consiglia vivamente di leggere le sezioni seguenti nella loro interezza prima di cominciare il processo di migrazione online.

La migrazione inizia quando si richiama l'operazione API `StartMigration` o il comando AWS CLI. Per la modalità cluster Redis disabilitata, il processo di migrazione rende il nodo primario del cluster ElastiCache per Redis una replica del nodo primario Redis di origine. Per la modalità cluster Redis abilitata, il processo di migrazione rende il nodo primario di ogni partizione ElastiCache una replica della partizione corrispondente del cluster di origine che include gli stessi slot.

Una volta pronte le modifiche lato client, richiamare l'operazione API `CompleteMigration`. Questa operazione API promuove la distribuzione ElastiCache a distribuzione Redis primaria con nodi di replica e primari (se applicabile). Ora è possibile reindirizzare l'applicazione client in modo che inizi a scrivere i dati su ElastiCache. Nel corso della migrazione, è possibile verificare lo stato della replica eseguendo il comando [redis-cli INFO](#) sui nodi Redis e sui nodi primari ElastiCache.

Fasi della migrazione

I seguenti argomenti delineano il processo di migrazione dei dati:

- [Preparazione dei nodi di fonte e di destinazione per la migrazione](#)
- [Test della migrazione dei dati](#)
- [Avvio della migrazione](#)
- [Verifica dell'avanzamento della migrazione dei dati](#)

- [Completamento della migrazione dei dati](#)

Preparazione dei nodi di fonte e di destinazione per la migrazione

È necessario assicurare che tutti e quattro i pre-requisiti menzionati di seguito siano soddisfatti prima di iniziare la migrazione dalla console ElastiCache, dall'API o dalla CLI AWS.

Per preparare i nodi di fonte e di destinazione Redis per la migrazione

1. Individuare l'implementazione ElastiCache di destinazione e accertarsi di potervi migrare i dati.

Un'implementazione ElastiCache esistente e appena creata deve soddisfare i seguenti requisiti per la migrazione:

- Usa il motore Redis versione 5.0.6 o successiva.
 - Non deve avere abilitata la crittografia in transito o inattiva.
 - Ha la funzione Multi-AZ abilitata.
 - Dispone di memoria sufficiente per l'adattamento dei dati del cluster Redis. Per configurare le impostazioni di memoria riservata corrette, consulta [Gestione della memoria prenotata](#).
 - Per la modalità cluster disabilitata, è possibile eseguire la migrazione direttamente dalle versioni Redis 2.8.21 e successive alla versione Redis 5.0.6 e successive utilizzando la CLI o le versioni Redis 5.0.6 e successive con la CLI o la console. Per la modalità cluster abilitata, è possibile eseguire la migrazione direttamente da qualsiasi versione Redis abilitata per la modalità cluster alla versione Redis 5.0.6 e successive utilizzando la CLI o le versioni Redis 5.0.6 e successive con la CLI o la console.
 - Il numero di partizioni nella sorgente e nel target corrisponde.
 - Non fa parte di un datastore globale.
 - La suddivisione dei dati su più livelli è disabilitata.
2. Assicurati che le configurazioni di Redis open source e dell'implementazione ElastiCache per Redis siano compatibili.

Come requisito minimo, tutte le seguenti implementazioni ElastiCache di destinazione devono essere compatibili con la configurazione Redis per la replica Redis:

- Il cluster Redis non deve avere Redis AUTH abilitato.
- La configurazione `protected-mode` Redis deve essere impostata su `no`.

- Se Redis ha una config `bind`, questa deve essere aggiornata per consentire le richieste dai nodi ElastiCache.
 - Il numero di database logici deve essere lo stesso sul nodo ElastiCache e sul cluster Redis. Questo valore viene impostato utilizzando `databases` nella configurazione Redis.
 - I comandi Redis che eseguono la modifica dei dati non devono essere rinominati per consentire la riuscita della replica dei dati. ad esempio `sync`, `psync`, `info`, `config`, `command` e `cluster`.
 - Per replicare i dati dal cluster Redis per ElastiCache, verificare che vi siano CPU e memoria sufficienti per gestire questo carico ulteriore. Questo carico proviene dal file RDB creato dal cluster Redis e trasferito tramite la rete al nodo ElastiCache.
 - Tutte le istanze Redis sul cluster di origine devono essere eseguite sulla stessa porta.
3. Verifica che l'istanza possa connettersi a ElastiCache procedendo come segue:
- Verifica che l'indirizzo IP di ogni istanza sia privato.
 - Assegnare o creare l'implementazione ElastiCache nello stesso virtual private cloud (VPC) di Redis sull'istanza (scelta consigliata).
 - Se i VPC sono differenti, configurare il peering VPC per consentire l'accesso tra i due nodi. Per ulteriori informazioni sul peering VPC, consulta [Modelli di accesso a una cache ElastiCache in un Amazon VPC](#).
 - Il gruppo di sicurezza collegato alle istanze Redis deve consentire il traffico in entrata dai nodi ElastiCache.
4. Verificare che l'applicazione possa indirizzare il traffico ai nodi ElastiCache quando la migrazione dei dati è completa. Per ulteriori informazioni, consulta [Modelli di accesso a una cache ElastiCache in un Amazon VPC](#).

Test della migrazione dei dati

Quando tutti i prerequisiti sono completi, è possibile convalidare la configurazione della migrazione utilizzando AWS Management Console, l'API ElastiCache o AWS CLI. L'esempio seguente mostra come utilizzare la CLI.

Puoi testare la migrazione utilizzando il comando `test-migration` con i seguenti parametri:

- `--replication-group-id` - L'ID del gruppo di replica a cui devono essere migrati i dati.

- `--customer-node-endpoint-list` - Elenco degli endpoint da cui migrare i dati. L'elenco deve contenere un solo elemento.

L'esempio seguente utilizza la CLI.

```
aws elasticache test-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

ElastiCache convaliderà la configurazione della migrazione senza alcuna migrazione effettiva dei dati.

Avvio della migrazione

Quando tutti i prerequisiti sono disponibili, è possibile avviare la migrazione dei dati utilizzando la AWS Management Console, l'API ElastiCache o lo AWS CLI. Se la modalità cluster è abilitata e la migrazione degli slot è diversa, verrà eseguito un resharding prima della migrazione in tempo reale. L'esempio seguente mostra come utilizzare la CLI.

Note

Si consiglia di utilizzare TestMigration API per convalidare la configurazione della migrazione. Tuttavia, questo è strettamente facoltativo.

Puoi avviare la migrazione utilizzando il comando `start-migration` con i seguenti parametri:

- `--replication-group-id` - L'identificativo del gruppo di replica ElastiCache di destinazione
- `--customer-node-endpoint-list` - Un elenco di endpoint con indirizzi DNS o IP e la porta in cui è in esecuzione il cluster Redis fonte. L'elenco può accettare solo un elemento sia per la modalità cluster disabilitata che per la modalità cluster abilitata. Se è stata abilitata la replica concatenata, l'endpoint può puntare a una replica piuttosto che al nodo master nel cluster Redis.

L'esempio seguente utilizza la CLI.

```
aws elasticache start-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

Quando si esegue questo comando, il nodo primario ElastiCache (in ogni partizione) si configura per diventare una replica dell'istanza Redis (nella partizione corrispondente che possiede gli stessi slot

nel cluster Redis abilitato). Lo stato del cluster ElastiCache passa a migrazione in corso e ha inizio la migrazione dei dati dall'istanza Redis al nodo primario. A seconda della dimensione dei dati e del carico sull'istanza Redis, il completamento della migrazione può richiedere del tempo. Nel corso della migrazione, è possibile verificare lo stato della replica eseguendo il comando [redis-cli INFO](#) sull'istanza Redis e sul nodo primario ElastiCache.

Se l'operazione di replica riesce, le scritture dei dati sulle istanze Redis si propagano al cluster ElastiCache. È possibile utilizzare i nodi ElastiCache per le letture. Tuttavia, non puoi scrivere sul cluster ElastiCache. Se il nodo primario ElastiCache ha altri nodi di replica collegati, questi nodi di replica continuano a replicarsi dal nodo primario ElastiCache. In questo modo, tutti i dati del cluster Redis vengono replicati in tutti i nodi del cluster ElastiCache.

Se il nodo primario ElastiCache non può diventare una replica dell'istanza Redis, effettua diversi nuovi tentativi prima di promuoversi nuovamente come master. Lo stato del cluster ElastiCache quindi passa a available (disponibile) e viene inviato un evento del gruppo di replica sull'errore all'inizio della migrazione. Per risolvere l'errore, verificare quanto segue:

- Andare all'evento del gruppo di replica. Utilizzare qualsiasi informazione specifica dell'evento per correggere l'errore di migrazione.
- Se l'evento non fornisce informazioni specifiche, verificare di aver seguito le linee guida in [Preparazione dei nodi di fonte e di destinazione per la migrazione](#).
- Verifica che la configurazione di routing per il VPC e per le sottoreti consenta il traffico tra i nodi ElastiCache e le istanze Redis.
- Verifica che il gruppo di sicurezza associato alle istanze Redis consenta il traffico in entrata dai nodi ElastiCache.
- Controlla i log Redis per le istanze Redis per ulteriori informazioni su errori specifici della replica.

Verifica dell'avanzamento della migrazione dei dati

Una volta iniziata la migrazione dei dati, è possibile seguirne l'avanzamento in questo modo:

- Verificare che `master_link_status` Redis sia up nel comando INFO sul nodo primario ElastiCache. È possibile trovare questa operazione nella console ElastiCache. Scegli il cluster e sotto Metriche CloudWatch, osserva Stato di integrità del collegamento principale). Una volta che il valore raggiunge 1, i dati sono sincronizzati.

- È possibile verificare che la replica ElastiCache abbia lo stato online eseguendo il comando `INFO` sulle istanze Redis. In questo modo è possibile ottenere informazioni sul rallentamento della replica.
- Verifica che il buffer di output del client sia basso utilizzando il comando Redis [CLIENT LIST](#) sulle istanze Redis.

Una volta completata la migrazione, i dati vengono sincronizzati con qualsiasi nuova scrittura in arrivo sul nodo primario del cluster Redis.

Completamento della migrazione dei dati

Una volta pronti a effettuare il trasferimento al cluster ElastiCache, utilizzare il comando CLI `complete-migration` con i seguenti parametri:

- `--replication-group-id` – L'identificatore del gruppo di replica.
- `--force` – Un valore che forza l'arresto della migrazione senza garantire la sincronizzazione dei dati.

Di seguito è riportato un esempio.

```
aws elasticache complete-migration --replication-group-id test-cluster
```

Quando si esegue questo comando, il nodo primario ElastiCache (in ogni partizione) interrompe la replica dall'istanza Redis e la promuove a primaria. Questa operazione in genere viene completata in cinque minuti. Per confermare la promozione a primaria, controllare l'evento `Complete Migration successful for test-cluster`. A questo punto, puoi indirizzare l'applicazione alle operazioni di scrittura e lettura ElastiCache. Lo stato del cluster ElastiCache dovrebbe cambiare da `migrating` (migrazione in corso) a `available` (disponibile).

Se la promozione a master non riesce, il nodo primario ElastiCache continua con la replica dall'istanza Redis. Il cluster ElastiCache continua a essere nello stato migrazione in corso e viene inviato un messaggio di evento del gruppo di replica sull'errore. Per risolvere questo errore, seguire questa procedura:

- Controllare l'evento del gruppo di replica. Utilizzare le informazioni specifiche dell'evento per correggere l'errore.

- Potrebbe essere generato un messaggio di evento sui dati non sincronizzati. In questo caso, verificare che ElastiCache primario possa replicare dall'istanza Redis e che entrambi siano sincronizzati. Se si desidera ancora arrestare la migrazione, è possibile eseguire il comando precedente con l'opzione `-force`.
- È possibile che venga visualizzato un messaggio di evento se uno dei nodi ElastiCache è in fase di sostituzione. È possibile riprovare l'intera fase di migrazione al termine della sostituzione.

Esecuzione della migrazione dei dati online utilizzando la console

Puoi utilizzare la AWS Management Console per migrare i dati dal tuo cluster al cluster Redis.

Per eseguire la migrazione dei dati online utilizzando la console

1. Accedere alla e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Creare un nuovo cluster Redis o scegliere un cluster esistente. Verificare che il cluster soddisfi i seguenti requisiti:
 - La versione del motore Redis deve essere almeno 5.0.6 o successiva.
 - Il cluster Redis non deve avere Redis AUTH abilitato.
 - La configurazione `protected-mode` Redis deve essere impostata su `no`.
 - Se Redis ha una config `bind`, questa deve essere aggiornata per consentire le richieste dai nodi ElastiCache.
 - Il numero di database deve essere lo stesso tra il nodo ElastiCache e il cluster Redis. Questo valore viene impostato utilizzando `databases` nella configurazione Redis.
 - I comandi Redis che eseguono la modifica dei dati non devono essere rinominati per consentire la riuscita della replica dei dati.
 - Per replicare i dati dal cluster Redis per ElastiCache, verificare che vi siano CPU e memoria sufficienti per gestire questo carico ulteriore. Questo carico proviene dal file RDB creato dal cluster Redis e trasferito tramite la rete al nodo ElastiCache.
 - Lo stato del cluster è `available` (disponibile).
3. Con il cluster selezionato, scegliere `Migrate Data from Endpoint` (Migra i dati dall'endpoint) per `Actions` (Operazioni).
4. Nella casella di dialogo Migrazione dei dati dall'endpoint immetti l'indirizzo IP e la porta in cui è disponibile il cluster Redis.

⚠ Important

L'indirizzo IP deve essere esatto. Se si immette l'indirizzo errato, la migrazione non riesce.

5. Scegliere Start Migration (Avvia migrazione).

Quando il cluster inizia la migrazione, il suo stato passa a Modifying (Modifica in corso), quindi a Migrating (Migrazione in corso).

6. Monitorare l'avanzamento della migrazione selezionando Events (Eventi) sul pannello di navigazione.

La migrazione può essere interrotta in qualsiasi punto del processo. Per questa operazione, scegliere il cluster, quindi Stop Data Migration (Arresta migrazione dati) per Actions (Operazioni). Lo stato del cluster diventa Available (Disponibile).

Se la migrazione riesce, lo stato del cluster diventa Available (Disponibile) e il registro eventi mostra quanto segue:

```
Migration operation succeeded for replication group ElastiCacheClusterName.
```

Se la migrazione non riesce, il cluster passa allo stato Available (Disponibile) e il registro eventi mostra quanto segue:

```
Migration operation failed for replication group ElastiCacheClusterName.
```

Scelta di regioni e zone di disponibilità

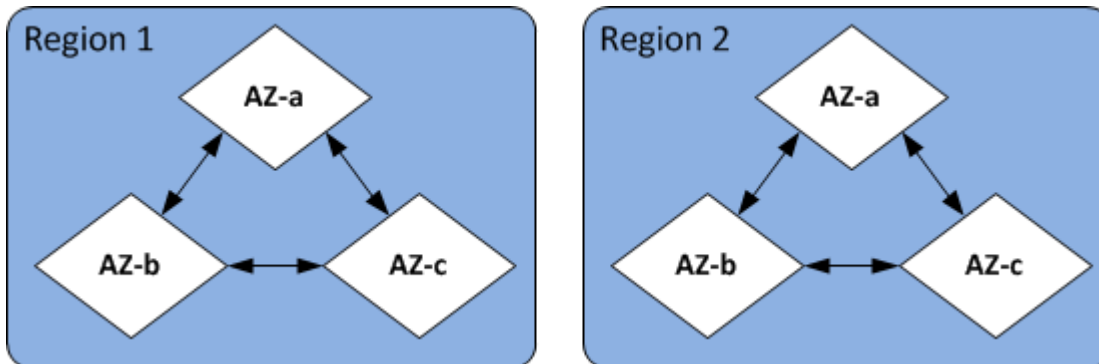
AWS Le risorse di cloud computing sono ospitate in strutture di data center ad alta disponibilità. Per fornire ulteriore scalabilità e affidabilità, queste strutture di data center sono in ubicazioni fisiche diverse. Tali località sono categorizzate in base a regioni e zone di disponibilità.

AWS Le regioni sono ampie e ampiamente distribuite in località geografiche separate. Le zone di disponibilità sono località distinte all'interno di una AWS regione progettate per essere isolate dai guasti in altre zone di disponibilità. Forniscono connettività di rete economica e a bassa latenza ad altre zone di disponibilità nella stessa regione. AWS

⚠ Important

Ciascuna regione è completamente indipendente. Qualsiasi ElastiCache attività 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](#).



Regioni e zone di disponibilità

Argomenti

- [Ubicazione dei nodi](#)
- [Regioni ed endpoint supportati](#)
- [Utilizzo di local zones con ElastiCache](#)
- [Utilizzo Outposts](#)

Ubicazione dei nodi

Amazon ElastiCache supporta la localizzazione di tutti i nodi di un cluster in una o più zone di disponibilità (AZ). Inoltre, se scegli di localizzare i tuoi nodi in più AZ (scelta consigliata), ti ElastiCache consente di scegliere le AZ per ogni nodo o di scegliere quelle ElastiCache per te.

Ubicando i nodi in diverse zone di disponibilità, elimini la possibilità che un errore, ad esempio un'interruzione di corrente, in una zona di disponibilità causi l'interruzione dell'intero sistema. L'esecuzione del test ha dimostrato che non esiste una differenza di latenza significativa sia se si posizionano tutti i nodi in una zona di disponibilità sia se si distribuiscono tra più zone di disponibilità.

Puoi specificare una zona di disponibilità per ciascun nodo quando crei un cluster o aggiungendo nodi quando modifichi un cluster esistente. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Creazione di un cluster](#)
- [Modifica di un cluster ElastiCache](#)
- [Aggiunta di nodi a un cluster](#)

Regioni ed endpoint supportati

Amazon ElastiCache è disponibile in più AWS regioni. Ciò significa che puoi avviare ElastiCache cluster in località che soddisfano i tuoi requisiti. Ad esempio, puoi lanciare il prodotto nella AWS regione più vicina ai tuoi clienti o farlo in una AWS regione particolare per soddisfare determinati requisiti legali.

Ogni regione è pensata per essere completamente isolata dalle altre regioni . All'interno di ciascuna regione ci sono più zone di disponibilità. ElastiCache Le cache serverless replicano automaticamente i dati su più zone di disponibilità (ad eccezione us-west-1 dei casi in cui i dati vengono replicati in due zone di disponibilità) per un'elevata disponibilità. Quando si progetta il proprio ElastiCache cluster, è possibile scegliere di avviare i nodi in diverse AZ per ottenere la tolleranza agli errori. Per ulteriori informazioni sulle regioni e sulle zone di disponibilità, consulta [Scelta di regioni e zone di disponibilità](#) all'inizio di questo argomento.

Regioni in cui ElastiCache è supportato

Nome regione/Regione	Endpoint	Protocollo	
Stati Uniti orientali (Ohio) us-east-2	elasticache.us-east-2.amazonaws.com	HTTPS	
Regione Stati Uniti orientali (Virginia settentrionale) us-east-1	elasticache.us-east-1.amazonaws.com	HTTPS	
Regione Stati Uniti occidentali (California settentrionale) us-west-1	elasticache.us-west-1.amazonaws.com	HTTPS	
Stati Uniti occidentali (Oregon) us-west-2	elasticache.us-west-2.amazonaws.com	HTTPS	
Regione Canada (Centrale) ca-central-1	elasticache.ca-central-1.amazonaws.com	HTTPS	
Regione Canada (Occidentale) ca-west-1	elasticache.ca-west-1.amazonaws.com	HTTPS	
Asia Pacifico (Giacarta) ap-southeast-3	elasticache.ap-southeast-3.amazonaws.com	HTTPS	

Nome regione/Regione	Endpoint	Protocollo	
Regione Asia Pacifico (Mumbai) ap-south-1	elasticache.ap-south-1.amazonaws.com	HTTPS	
Regione Asia Pacifico (Hyderabad) ap-south-2	elasticache.ap-south-2.amazonaws.com	HTTPS	
Regione Asia Pacifico (Tokyo) ap-northeast-1	elasticache.ap-northeast-1.amazonaws.com	HTTPS	
Regione Asia Pacifico (Seoul) ap-northeast-2	elasticache.ap-northeast-2.amazonaws.com	HTTPS	
Regione Asia Pacifico (Osaka-Locale) ap-northeast-3	elasticache.ap-northeast-3.amazonaws.com	HTTPS	
Regione Asia Pacifico (Singapore) ap-southeast-1	elasticache.ap-southeast-1.amazonaws.com	HTTPS	
Asia Pacifico (Sydney) ap-southeast-2	elasticache.ap-southeast-2.amazonaws.com	HTTPS	

Nome regione/Regione	Endpoint	Protocollo	
Regione Europa (Francoforte) eu-central-1	elasticache.eu-central-1.amazonaws.com	HTTPS	
Regione Europa (Zurigo) eu-central-2	elasticache.eu-central-2.amazonaws.com	HTTPS	
Regione Europa (Stoccolma) eu-north-1	elasticache.eu-north-1.amazonaws.com	HTTPS	
Regione Medio Oriente (Bahrein) me-south-1	elasticache.me-south-1.amazonaws.com	HTTPS	
Regione Medio Oriente (Emirati Arabi Uniti) me-central-1	elasticache.me-central-1.amazonaws.com	HTTPS	
Europa (Irlanda) eu-west-1	elasticache.eu-west-1.amazonaws.com	HTTPS	
Regione Europa (Londra) eu-west-2	elasticache.eu-west-2.amazonaws.com	HTTPS	

Nome regione/Regione	Endpoint	Protocollo	
Regione UE (Parigi) eu-west-3	elasticache.eu-west-3.amazonaws.com	HTTPS	
Regione Europa (Milano) eu-south-1	elasticache.eu-south-1.amazonaws.com	HTTPS	
Regione Europa (Spagna) eu-south-2	elasticache.eu-south-2.amazonaws.com	HTTPS	
Regione Sud America (San Paolo) sa-east-1	elasticache.sa-east-1.amazonaws.com	HTTPS	
Regione Cina (Pechino) cn-north-1	elasticache.cn-north-1.amazonaws.com.cn	HTTPS	
Regione Cina (Ningxia) cn-northwest-1	elasticache.cn-northwest-1.amazonaws.com.cn	HTTPS	
Regione Asia Pacifico (Hong Kong) ap-east-1	elasticache.ap-east-1.amazonaws.com	HTTPS	

Nome regione/Regione	Endpoint	Protocollo
Regione Africa (Città del Capo) af-south-1	elasticache.af-south-1.amazonaws.com	HTTPS
Regione di Israele (Tel Aviv) il-central-1	elasticache.il-central-1.amazonaws.com	HTTPS
AWS GovCloud (Stati Uniti occidentali) us-gov-west-1	elasticache.us-gov-west-1.amazonaws.com	HTTPS
AWS GovCloud (Stati Uniti orientali) us-gov-east-1	elasticache.us-gov-east-1.amazonaws.com	HTTPS

Per informazioni sull'utilizzo di AWS GovCloud (Stati Uniti) con ElastiCache, vedi [Servizi nella regione AWS GovCloud \(Stati Uniti\): ElastiCache](#).

Alcune regioni supportano un set secondario di tipi di nodo. Per una tabella dei tipi di nodi supportati per AWS regione, vedere [Tipi di nodi supportati in base alla regione AWS](#).

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

Utilizzo di local zones con ElastiCache

Una zona locale è un'estensione di una regione AWS che è geograficamente vicina agli utenti. Puoi estendere qualsiasi servizio virtual private cloud (VPC) dalla regione AWS principale a Local Zones tramite la creazione di una nuova sottorete da assegnare alla zona locale. Quando si crea una sottorete in una zona locale, il VPC viene esteso anche a tale zona locale. La sottorete nell'area locale funziona allo stesso modo delle altre sottoreti nel VPC.

Utilizzando Local Zones è possibile posizionare risorse come un cluster ElastiCache in più posizioni vicine agli utenti.

Quando si crea un cluster ElastiCache, è possibile scegliere una sottorete in una zona locale. Le Local Zones hanno le loro connessioni a Internet e supportano AWS Direct Connect. Pertanto, le risorse create in un'area locale possono servire gli utenti locali con comunicazioni a latenza molto bassa. Per ulteriori informazioni, consulta [AWS Local Zones](#).

Una zona locale è rappresentata da un codice regione AWS seguito da un identificatore che indica la posizione, ad esempio `us-west-2-lax-1a`.

Al momento, le Local Zones disponibili sono `us-west-2-lax-1a` e `us-west-2-lax-1b`.

Le seguenti limitazioni si applicano a ElastiCache per Local Zones:

- I datastore globali non sono supportati.
- La migrazione online non è supportata.
- I seguenti tipi di nodi sono supportati dalle Local Zones:
 - Generazione attuale:

Tipi di nodi M5: `cache.m5.large`, `cache.m5.xlarge`, `cache.m5.2xlarge`,
`cache.m5.4xlarge`, `cache.m5.12xlarge`, `cache.m5.24xlarge`

Tipi di nodi R5: `cache.r5.large`, `cache.r5.xlarge`, `cache.r5.2xlarge`,
`cache.r5.4xlarge`, `cache.r5.12xlarge`, `cache.r5.24xlarge`

Tipi di nodo T3: `cache.t3.micro`, `cache.t3.small`, `cache.t3.medium`

Abilitazione di una zona locale

1. Attivare la zona locale nella console di Amazon EC2.

Per ulteriori informazioni, vedere [Abilitazione di Local Zones](#) nella Guida per l'utente di Amazon EC2 per le istanze Linux.

2. Creare una sottorete nella zona locale.

Per ulteriori informazioni, consulta [Creazione di una sottorete nel VPC](#) nella Guida per l'utente di Amazon VPC.

3. Creare un gruppo di sottoreti di ElastiCache in Local Zone.

Quando si crea un gruppo di sottoreti di ElastiCache, scegliere il gruppo Zona di disponibilità per la zona locale.

Per ulteriori informazioni, consulta [Creazione di un gruppo di sottoreti](#) nella guida per l'utente ElastiCache.

4. Creare un cluster ElastiCache for Redis che utilizza la sottorete ElasticaCache nella zona locale. Per ulteriori informazioni, consulta uno dei seguenti argomenti:
 - [Creazione di un cluster Redis \(modalità cluster disabilitata\) \(Console\)](#)
 - [Creazione di un cluster Redis \(modalità cluster abilitata\) \(console\)](#).

Utilizzo Outposts

AWS Outposts è un servizio completamente gestito che estende l'infrastruttura AWS, i servizi, le API e gli strumenti alla sede del cliente. Fornendo l'accesso locale all'infrastruttura gestita AWS, AWSOutposts consente ai clienti di creare ed eseguire applicazioni in locale utilizzando le stesse interfacce di programmazione delle regioni AWS, utilizzando al contempo le risorse di calcolo e archiviazione locali per le esigenze di bassa latenza e minore elaborazione dei dati locali. Un Outpost è un pool di capacità di AWS elaborazione e storage distribuito presso la sede del cliente. AWS gestisce, monitora e gestisce questa capacità come parte di una regione AWS. Puoi creare sottoreti nell'Outpost e specificarle quando creiAWSrisorse quali cluster ElastiCache.

Note

In questa versione si applicano le seguenti limitazioni:

- ElastiCache per Outposts supporta solo le famiglie di nodi M5 e R5.
- La migrazione live non è supportata.
- Multi-AZ (la replica cross-outpost non è supportata).
- Gli snapshot locali non sono supportati.
- ElastiCache per gli Outposts non è supportato nelle seguenti aree: cn-northwest-1, cn-northwest-1 e ap-northeast-3.

Utilizzo degli Outposts con la console Redis

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegliere Redis.
3. Nello stato Motore cluster, selezionare Redis.
4. In Posizione, scegliere On-Premise: crea le tue istanze ElastiCache su AWS Outposts.

Configurare le opzioni locali

È possibile scegliere un Outpost disponibile per aggiungere il cluster di cache oppure, se non ci sono Outposts disponibili, crearne uno nuovo seguendo la procedura seguente:

In Opzioni on-premise:

1. In Impostazioni Redis:
 - a. In Name (Nome), immettere un nome per il cluster Redis.
 - b. Descrizione cluster: immettere una descrizione per il cluster Redis.
 - c. Compatibilità della versione del motore.: la versione del motore è basata sull'AWS Regione dell'Avampost
 - d. In Port (Porta), accettare la porta di default, 6379. Se esiste un valido motivo per utilizzare una porta diversa, digitare il numero della porta.
 - e. Gruppo di parametri: Utilizzare il menu a discesa per scegliere un gruppo di parametri di default o personalizzato.
 - f. Tipo di nodo: le istanze disponibili si basano sulla disponibilità degli Outposts. Assistente porting per .NET per Outposts supporta solo le famiglie di nodi M5 e R5. Dall'elenco a discesa, scegliere Outposts e quindi scegliere un tipo di nodo disponibile che si desidera utilizzare per questo cluster. Quindi, scegliere Salva.
 - g. Numero di repliche: immettere il numero di repliche di lettura desiderate creare per questo gruppo di replica. È necessario che sia presente almeno una e non più di cinque repliche di lettura. Il valore di default è 2.

I nomi generati automaticamente delle repliche di lettura seguono lo stesso schema del nome del cluster primario, con un trattino e un numero sequenziale di tre cifre aggiunti alla fine, iniziando con -002. Ad esempio,

se il gruppo di replica è denominato `MyGroup`, allora i nomi dei secondari sarebbero `MyGroup-002`, `MyGroup-003`, `MyGroup-004`, `MyGroup-005`, `MyGroup-006`.

2. In Impostazioni avanzate di Redis:

a. Gruppo di sottoreti: Dall'elenco, seleziona `Crea nuova`.

- Nome: Inserire un nome breve per il gruppo di sottoreti.
- Descrizione: Inserire una descrizione del gruppo di sottoreti.
- ID VPC: l'ID VPC deve corrispondere al VPC dell'Avamposto. Se si seleziona un VPC senza ID sottorete nei Outposts, l'elenco verrà restituito vuoto.
- Zona di disponibilità o Availability: seleziona l'Avamposto che stai utilizzando.
- ID sottorete: scegliere un ID di sottorete disponibile per l'Avamposto. Se non sono disponibili ID di sottoreti, devi crearli. Per ulteriori dettagli, vedi [Crea una sottorete](#).

b. Selezionare `Crea`.

Visualizzazione dei dettagli del cluster Outpost

Nella pagina elenco Redis scegliere un cluster appartenente a un AWS Avamposto e annotare quanto segue quando si visualizza il `Dettagli del cluster`:

- Availability zone (Zona di disponibilità): questo rappresenterà l'Avamposto, utilizzando un ARN (Nome risorsa Amazon) e il `AWSNumero` risorsa.
- Nome Avampost: il nome dell'AWS Avamposto.

Utilizzo degli Outposts con il CLI AWS

È possibile utilizzare AWS Command Line Interface (AWS CLI) per controllare più servizi AWS dalla riga di comando e automatizzarli tramite script. È possibile utilizzare la AWS (CLI) per operazioni ad hoc (una tantum).

Download e configurazione dell' AWS CLI

L'AWS CLI funziona in Windows, macOS o Linux. Utilizzare la procedura seguente per scaricarla e configurarla.

Per scaricare, installare e configurare la CLI

1. Scaricare la CLI AWS dalla pagina Web [AWSInterfaccia a riga di comando](#).

2. Seguire le istruzioni per [l'installazione della CLI AWS](#) e la [configurazione della AWS CLI](#) nella AWS Command Line Interface guida per l'utente.

Utilizzo di AWS CLI con Outposts

Utilizzare la seguente operazione CLI per creare un cluster di cache che utilizza Outposts:

- [create-cache-cluster](#)— Utilizzando questa operazione, il `outpost-mode` accetta un valore che specifica se i nodi nel cluster di cache vengono creati in un singolo Outpost o su più Outposts.

Note

Al momento, solo `single-outpost` Modalità supportata.

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id cache cluster id \  
  --outpost-mode single-outpost \  
  \
```

Lavorare con ElastiCache

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

Argomenti

- [Snapshot e ripristino](#)
- [Versioni e aggiornamenti del motore](#)
- [Best practice e strategie di caching ElastiCache](#)
- [Gestione del cluster progettato autonomamente](#)
- [Scalabilità per Redis ElastiCache](#)
- [Nozioni di base di JSON in ElastiCache per Redis](#)
- [Assegnazione di tag delle risorse ElastiCache](#)
- [Utilizzo di Amazon ElastiCache Well-Architected Lens](#)
- [Procedure e best practice comuni per la risoluzione dei problemi](#)
- [Ulteriori passaggi per la risoluzione dei problemi](#)

Snapshot e ripristino

Le cache Amazon ElastiCache che eseguono Redis possono effettuare il backup dei dati creando uno snapshot. Il backup può essere utilizzato per ripristinare una cache o inizializzare i dati in una nuova cache. Il backup è costituito dai metadati della cache, insieme a tutti i dati presenti nella cache. Tutti i backup vengono scritti su Amazon Simple Storage Service (Amazon S3), che fornisce uno storage durevole. In qualsiasi momento, puoi ripristinare i dati creando una nuova cache Redis e popolandola con i dati di un backup. Con ElastiCache, puoi gestire i backup utilizzando lo AWS Management Console, il AWS Command Line Interface(AWS CLI) e l'API ElastiCache.

Se pianifichi di eliminare una cache ed è importante preservare i dati, puoi prendere precauzioni aggiuntive. Per farlo, crea innanzitutto un backup manuale, verifica che lo stato sia disponibile, quindi elimina la cache. In questo modo hai la garanzia che, in caso di errore del backup, disporrai comunque dei dati della cache. Puoi provare nuovamente a creare un backup, seguendo le best practice illustrate in precedenza.

Argomenti

- [Vincoli del backup](#)
- [Impatto sulle prestazioni dei backup di cluster progettati autonomamente](#)
- [Pianificazione di backup automatici](#)
- [Esecuzione di backup manuali](#)
- [Creazione di un backup finale](#)
- [Descrizione dei backup](#)
- [Copia di backup](#)
- [Esportazione di un backup](#)
- [Ripristino da un backup in una nuova cache](#)
- [Eliminazione di un backup](#)
- [Assegnazione di tag di backup](#)
- [Inizializzazione di un nuovo cluster progettato autonomamente con un backup creato esternamente](#)

Vincoli del backup

Durante la pianificazione o la creazione di backup occorre considerare i seguenti vincoli:

- Al momento, backup e ripristino sono supportati solo per cache in esecuzione su Redis.
- Per i cluster , backup e ripristino non sono supportati sui nodi cache `.t1.micro`. Tutti gli altri tipi di nodi di cache sono supportati.
- Per cluster Redis, backup e ripristino sono supportati per tutti i tipi di nodi.
- Durante qualsiasi periodo continuo di 24 ore, non puoi creare più di 20 backup manuali per nodo nel cluster.
- Redis (modalità cluster abilitata) supporta solo l'esecuzione di backup a livello di cluster (per l'API o la CLI, il livello del gruppo di replica). Redis (modalità cluster abilitata) non supporta i backup a livello di shard (per l'API o la CLI, il livello del gruppo di nodi).
- Durante il processo di backup non puoi eseguire operazioni API o CLI aggiuntive sul cluster.
- Se si utilizzano cluster con tiering di dati, non è possibile esportare un backup in Amazon S3.
- È possibile ripristinare un backup di un cluster utilizzando il tipo di nodo `r6gd` solo su cluster che utilizzano il tipo di nodo `r6gd`.

Impatto sulle prestazioni dei backup di cluster progettati autonomamente

I backup sulle cache serverless sono trasparenti per l'applicazione senza alcun impatto sulle prestazioni. Tuttavia, quando si creano backup per cluster progettati autonomamente, può esserci un certo impatto sulle prestazioni a seconda della memoria riservata disponibile.

Di seguito sono elencate le linee guida per migliorare le prestazioni di backup dei cluster progettati autonomamente.

- Impostare il parametro `reserved-memory-percent` : Per mitigare la paginazione eccessiva, ti consigliamo di impostare il parametro `reserved-memory-percent`. Questo parametro impedisce a Redis di consumare tutta la memoria disponibile del nodo e consente di ridurre la quantità di paginazione. È anche possibile che si verifichino miglioramenti delle prestazioni utilizzando semplicemente un nodo più grande. Per ulteriori informazioni sui parametri `reserved-memory` e `reserved-memory-percent`, consulta [Gestione della memoria prenotata](#).
- Creare backup da una replica di lettura : Se si esegue Redis in un gruppo di nodi con più di un nodo, puoi accettare un backup dal nodo primario o da una delle repliche di lettura. A causa delle risorse di sistema richieste durante BGSAVE, ti consigliamo di creare backup da una delle repliche di lettura. Nel corso della creazione del backup dalla replica, il nodo primario non viene alterato dai requisiti della risorsa BGSAVE. Il nodo primario può continuare a servire le richieste senza rallentamenti.

A tale scopo, consulta [Creazione di un backup manuale \(Console\)](#) e nel Nome del cluster nella finestra Crea backup, scegliere una replica anziché il nodo primario di default.

Se elimini un gruppo di replica e richiedi un backup finale, ElastiCache prenderà sempre il backup dal nodo primario. Questo consente di acquisire i dati Redis più recenti prima che il gruppo di replica venga eliminato.

Pianificazione di backup automatici

Puoi abilitare i backup automatici per qualsiasi cache ElastiCache per Redis. Quando i backup automatici sono abilitati, ElastiCache crea quotidianamente un backup della cache. Non vi è alcun impatto sulla cache e la modifica è immediata. I backup automatici possono fornire protezione da perdita di dati. In caso di errore, puoi creare una nuova cache ripristinando i dati dal backup più recente. Il risultato è una cache avviata a caldo, precaricata con i dati e pronta per l'uso. Per ulteriori informazioni, consulta [Ripristino da un backup in una nuova cache](#).

Quando pianifichi backup automatici, considera le seguenti impostazioni:

- **Ora di inizio del backup:** l'ora in cui ElastiCache inizia a creare un backup. Puoi impostare la finestra di backup in qualsiasi momento e quando è più comodo. Se non specifichi una finestra di backup, ElastiCache ne assegna automaticamente una.
- **Periodo di retention di backup :** Il numero di giorni in cui il backup viene mantenuto in Amazon S3. Ad esempio, se imposti il limite del periodo di retention su 5, un backup acquisito oggi verrà mantenuto per 5 giorni. Quando il periodo di retention scade, il backup viene automaticamente eliminato.

Il periodo di retention di backup massimo è 35 giorni. Se il periodo di conservazione dei backup è impostato su 0, i backup automatici sono disabilitati per la cache.

Puoi abilitare o disabilitare i backup automatici quando crei una nuova cache o aggiorni una cache Redis esistente mediante la console ElastiCache, la AWS CLI o l'API ElastiCache selezionando la casella **Abilita backup automatici** nella sezione **Impostazioni avanzate di Redis**.

Esecuzione di backup manuali

Oltre ai backup automatici, puoi creare un backup manuale in qualsiasi momento. A differenza dei backup automatici, che vengono eliminati automaticamente dopo un periodo di retention specificato, i backup manuali non dispongono di un periodo di retention dopo il quale vengono eliminati automaticamente. Anche se elimini la cache, tutti i backup manuali della cache vengono mantenuti. Se non desideri più mantenere un backup manuale, devi eliminarlo in maniera esplicita.

Oltre che direttamente, puoi creare un backup manuale in uno dei modi seguenti:

- [Copia di backup](#). Non importa se il backup di origine è stato creato automaticamente o manualmente.
- [Creazione di un backup finale](#). Crea immediatamente un backup prima di eliminare un cluster o un nodo.

Puoi creare un backup manuale di una cache utilizzando la AWS Management Console, la AWS CLI o l'API ElastiCache.

Creazione di un backup manuale (Console)

Per creare un backup di una cache (console)

1. Accedi a AWS Management Console e apri la console Amazon EC2 all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Dal pannello di navigazione scegli Cache Redis.
3. Scegli la casella a sinistra del nome del cluster Redis di cui eseguire il backup.
4. Scegli Backup.
5. Nella finestra di dialogo Crea backup, digita un nome per il backup nella casella Nome backup. Il nome deve indicare quale cluster è stato sottoposto a backup e la data e l'ora di esecuzione del backup.

I vincoli di denominazione del 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 posso terminare con un trattino.

6. Scegli Crea backup.

Lo stato de cluster cambia in creazione di snapshot.

Creazione di un backup manuale (AWS CLI)

Backup manuale di una cache serverless con la AWS CLI

Per creare un backup manuale di una cache mediante la AWS CLI, utilizza l'operazione `create-serverless-snapshot` della AWS CLI con i seguenti parametri:

- `--serverless-cache-name`: il nome della cache serverless di cui esegui il backup.
- `--serverless-cache-snapshot-name` - Nome dello snapshot da creare.

Per Linux, macOS o Unix:

- ```
aws elasticache create-serverless-snapshot \
 --serverless-cache-name CacheName \
 --serverless-cache-snapshot-name bkup-20231127
```

Per Windows:

- ```
aws elasticache create-serverless-snapshot ^  
    --serverless-cache-name CacheName ^  
    --serverless-cache-snapshot-name bkup-20231127
```

Backup manuale di un cluster progettato autonomamente con la AWS CLI

Per creare un backup manuale di un cluster mediante la AWS CLI, utilizza l'operazione `create-snapshot` della AWS CLI con i seguenti parametri:

- `--cache-cluster-id`
 - Se il cluster di cui stai eseguendo il backup non dispone di nodi di replica, `--cache-cluster-id` è il nome del cluster di cui stai eseguendo il backup, ad esempio *mycluster*.
 - Se il cluster di cui stai eseguendo il backup dispone di uno o più nodi di replica, `--cache-cluster-id` è il nome del nodo nel cluster che desideri utilizzare per il backup. Ad esempio, il nome potrebbe essere *mycluster-002*.

Utilizza questo parametro solo quando esegui il backup di un cluster Redis (modalità cluster disabilitata).

- `--replication-group-id` - Nome del cluster Redis (modalità cluster abilitata) (CLI/API: un gruppo di replica) da utilizzare come la fonte per il backup. Utilizza questo parametro quando esegui il backup di un cluster Redis (modalità cluster abilitata).
- `--snapshot-name` - Nome dello snapshot da creare.

I vincoli di denominazione del 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.

Esempio 1: Backup di un cluster Redis (modalità cluster disabilitata) senza nodi di replica

L'operazione AWS CLI seguente crea un backup `bkup-20150515` dal cluster Redis (modalità cluster disabilitata) `myNonClusteredRedis` che non dispone di repliche di lettura.

Per Linux, macOS o Unix:

```
aws elasticache create-snapshot \  
  --cache-cluster-id myNonClusteredRedis \  
  --snapshot-name bkup-20150515
```

Per Windows:

```
aws elasticache create-snapshot ^  
  --cache-cluster-id myNonClusteredRedis ^  
  --snapshot-name bkup-20150515
```

Esempio 2: Backup di un cluster Redis (modalità cluster disabilitata) con nodi di replica

La seguente operazione AWS CLI crea il backup `bkup-20150515` dal cluster Redis (modalità cluster disabilitata) `myNonClusteredRedis`. Questo backup ha una o più repliche di lettura.

Per Linux, macOS o Unix:

```
aws elasticache create-snapshot \  
  --cache-cluster-id myNonClusteredRedis-001 \  
  --snapshot-name bkup-20150515
```

Per Windows:

```
aws elasticache create-snapshot ^  
  --cache-cluster-id myNonClusteredRedis-001 ^  
  --snapshot-name bkup-20150515
```

Output di esempio: backup di un cluster Redis (modalità cluster disabilitata) con nodi di replica

L'aspetto dell'output dell'operazione è simile al seguente.

```
{  
  "Snapshot": {  
    "Engine": "redis",  
    "CacheParameterGroupName": "default.redis6.x",  
    "VpcId": "vpc-91280df6",  
    "CacheClusterId": "myNonClusteredRedis-001",  
    "SnapshotRetentionLimit": 0,  
    "NumCacheNodes": 1,  
    "SnapshotName": "bkup-20150515",  
    "CacheClusterCreateTime": "2017-01-12T18:59:48.048Z",  
    "AutoMinorVersionUpgrade": true,  
    "PreferredAvailabilityZone": "us-east-1c",  
    "SnapshotStatus": "creating",  
    "SnapshotSource": "manual",  
    "SnapshotWindow": "08:30-09:30",  
    "EngineVersion": "6.0",  
    "NodeSnapshots": [  
      {  
        "CacheSize": "",  
        "CacheNodeId": "0001",  
        "CacheNodeCreateTime": "2017-01-12T18:59:48.048Z"  
      }  
    ],  
    "CacheSubnetGroupName": "default",  
    "Port": 6379,  
    "PreferredMaintenanceWindow": "wed:07:30-wed:08:30",
```

```
    "CacheNodeType": "cache.m3.2xlarge",
    "DataTiering": "disabled"
  }
}
```

Esempio 3: Backup di un cluster per Redis (Modalità cluster abilitata)

La seguente operazione AWS CLI crea il backup `bkup-20150515` al cluster Redis (modalità cluster abilitata) `myClusteredRedis`. Nota l'uso di `--replication-group-id` anziché `--cache-cluster-id` per identificare l'fonte.

Per Linux, macOS o Unix:

```
aws elasticache create-snapshot \  
  --replication-group-id myClusteredRedis \  
  --snapshot-name bkup-20150515
```

Per Windows:

```
aws elasticache create-snapshot ^  
  --replication-group-id myClusteredRedis ^  
  --snapshot-name bkup-20150515
```

Output di esempio: backup di un cluster Redis (modalità cluster abilitata)

L'aspetto dell'output di questa operazione è simile al seguente.

```
{  
  "Snapshot": {  
    "Engine": "redis",  
    "CacheParameterGroupName": "default.redis6.x.cluster.on",  
    "VpcId": "vpc-91280df6",  
    "NodeSnapshots": [  
      {  
        "CacheSize": "",  
        "NodeGroupId": "0001"  
      },  
      {  
        "CacheSize": "",  
        "NodeGroupId": "0002"  
      }  
    ],  
  },  
}
```

```
"NumNodeGroups": 2,  
"SnapshotName": "bkup-20150515",  
"ReplicationGroupId": "myClusteredRedis",  
"AutoMinorVersionUpgrade": true,  
"SnapshotRetentionLimit": 1,  
"AutomaticFailover": "enabled",  
"SnapshotStatus": "creating",  
"SnapshotSource": "manual",  
"SnapshotWindow": "10:00-11:00",  
"EngineVersion": "6.0",  
"CacheSubnetGroupName": "default",  
"ReplicationGroupDescription": "2 shards 2 nodes each",  
"Port": 6379,  
"PreferredMaintenanceWindow": "sat:03:30-sat:04:30",  
"CacheNodeType": "cache.r3.large",  
"DataTiering": "disabled"  
}  
}
```

Argomenti correlati

Per ulteriori informazioni, consulta [create-snapshot](#) nella Guida di riferimento ai comandi della AWS CLI.

Creazione di un backup finale

Puoi creare un backup finale utilizzando la console ElastiCache, lo AWS CLI o l'API ElastiCache.

Creazione di un backup finale (console)

Puoi creare un backup finale quando elimini una cache Redis utilizzando la console ElastiCache.

Per creare un backup finale durante l'eliminazione di una cache Redis, nella finestra di dialogo di eliminazione scegli Sì per Crea backup e assegna un nome al backup.

Argomenti correlati

- [Utilizzo di AWS Management Console](#)
- [Eliminazione di un gruppo di replica \(console\)](#)

Creazione di un backup finale (AWS CLI)

Puoi creare un backup finale durante l'eliminazione di una cache Redis utilizzando la AWS CLI.

Argomenti

- [Eliminazione di una cache serverless Redis](#)
- [Durante l'eliminazione di un cluster Redis senza repliche di lettura](#)
- [Durante l'eliminazione di un cluster Redis con repliche di lettura](#)

Eliminazione di una cache serverless Redis

Per creare un backup finale, utilizza l'operazione AWS CLI di `delete-serverless-cache` con i seguenti parametri.

- `--serverless-cache-name`: nome della cache da eliminare.
- `--final-snapshot-name` : Nome del backup.

Il codice seguente consente di creare il backup finale `bkup-20231127-final` durante l'eliminazione della cache `myserverlesscache`.

Per Linux, macOS o Unix:

```
aws elasticache delete-serverless-cache \
```



```
--serverless-cache-name myserverlesscache \  
--final-snapshot-name bkup-20231127-final
```

Per Windows:

```
aws elasticache delete-serverless-cache ^  
--serverless-cache-name myserverlesscache ^  
--final-snapshot-name bkup-20231127-final
```

Per ulteriori informazioni, consulta [delete-serverless-cache](#) nella Guida di riferimento dei comandi di AWS CLI.

Durante l'eliminazione di un cluster Redis senza repliche di lettura

Per creare un backup finale, utilizza l'operazione AWS CLI di `delete-cache-cluster` con i seguenti parametri.

- `--cache-cluster-id` : Nome del cluster in corso di eliminazione.
- `--final-snapshot-identifier` : Nome del backup.

Il codice seguente consente di creare il backup finale `bkup-20150515-final` durante l'eliminazione del cluster `myRedisCluster`.

Per Linux, macOS o Unix:

```
aws elasticache delete-cache-cluster \  
--cache-cluster-id myRedisCluster \  
--final-snapshot-identifier bkup-20150515-final
```

Per Windows:

```
aws elasticache delete-cache-cluster ^  
--cache-cluster-id myRedisCluster ^  
--final-snapshot-identifier bkup-20150515-final
```

Per ulteriori informazioni, vedere [delete-cache cluster](#) nel AWS CLIRiferimento del comando.

Durante l'eliminazione di un cluster Redis con repliche di lettura

Per creare un backup finale durante l'eliminazione di un gruppo di replica, utilizza l'operazione AWS CLI di `delete-replication-group` con i seguenti parametri:

- `--replication-group-id` : Nome del gruppo di replica in corso di eliminazione.
- `--final-snapshot-identifier` : Nome del backup finale.

Il codice seguente prende il backup finale `bkup-20150515-final` durante l'eliminazione del gruppo di replica `myReplGroup`.

Per Linux, macOS o Unix:

```
aws elasticache delete-replication-group \  
  --replication-group-id myReplGroup \  
  --final-snapshot-identifier bkup-20150515-final
```

Per Windows:

```
aws elasticache delete-replication-group ^  
  --replication-group-id myReplGroup ^  
  --final-snapshot-identifier bkup-20150515-final
```

Per ulteriori informazioni, consulta [delete-replication-group \(elimina gruppo di replica\)](#) nel AWS CLI Riferimento del comando.

Descrizione dei backup

Le procedure seguenti illustrano come visualizzare un elenco dei backup. Se lo desideri, puoi visualizzare anche i dettagli di un particolare backup.

Descrizione dei backup (console)

Per visualizzare i backup mediante AWS Management Console

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione, scegliere Backups (Backup).
3. Per visualizzare i dettagli di un particolare backup, scegliere la casella a sinistra del nome del backup.

Descrizione dei backup serverless (AWS CLI)

Per visualizzare un elenco di backup serverless e opzionalmente i dettagli relativi a un backup specifico, utilizza l'operazione `describe-serverless-cache-snapshots` della CLI.

Examples (Esempi)

L'operazione seguente utilizza il parametro `--max-records` per elencare un massimo di 20 backup associati all'account. Se si omette il parametro `--max-records` vengono elencati fino a 50 backup.

```
aws elasticache describe-serverless-cache-snapshots --max-records 20
```

L'operazione seguente utilizza il parametro `--serverless-cache-name` per elencare solo i backup associati alla cache `my-cache`.

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-name my-cache
```

L'operazione seguente utilizza il parametro `--serverless-cache-snapshot-name` per visualizzare i dettagli del backup `my-backup`.

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-snapshot-name my-backup
```

Per ulteriori informazioni, consulta [describe-serverless-cache-snapshots](#) nella Guida di riferimento dei comandi di AWS CLI.

Descrizione dei backup di cluster progettati autonomamente (AWS CLI)

Per visualizzare un elenco di backup di cluster progettati autonomamente e opzionalmente i dettagli relativi a un backup specifico, utilizza l'operazione `describe-snapshots` della CLI.

Examples (Esempi)

L'operazione seguente utilizza il parametro `--max-records` per elencare un massimo di 20 backup associati all'account. Se si omette il parametro `--max-records` vengono elencati fino a 50 backup.

```
aws elasticache describe-snapshots --max-records 20
```

L'operazione seguente utilizza il parametro `--cache-cluster-id` per elencare solo i backup associati al cluster `my-cluster`.

```
aws elasticache describe-snapshots --cache-cluster-id my-cluster
```

L'operazione seguente utilizza il parametro `--snapshot-name` per visualizzare i dettagli del backup `my-backup`.

```
aws elasticache describe-snapshots --snapshot-name my-backup
```

Per ulteriori informazioni, consulta [describe-snapshot](#) nel AWS CLI Guida di riferimento dei comandi.

Copia di backup

Puoi creare una copia di qualsiasi backup, a prescindere che sia stato eseguito automaticamente o manualmente. Puoi anche esportare il backup per consentire di accedervi dall'esterno di ElastiCache. Per ulteriori informazioni sull'esportazione del backup, consulta [Esportazione di un backup](#).

I passaggi seguenti illustrano come copiare un backup.

Copia dei backup (console)

Copia di un backup (console)

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Per visualizzare un elenco dei backup, dal pannello di navigazione, scegliere Backups (Backup).
3. Dall'elenco di backup, scegliere la casella a sinistra del nome del backup che si desidera copiare.
4. Seleziona Operazioni e quindi Copia.
5. Nella casella New backup name (Nome del nuovo backup), digitare un nome per il nuovo backup.
6. Scegli Copia.

Copia di un backup serverless (AWS CLI)

Per copiare un backup di una cache serverless utilizza l'operazione `copy-serverless-cache-snapshot`.

Parametri

- `--source-serverless-cache-snapshot-name` - Nome del backup da copiare.
- `--target-serverless-cache-snapshot-name` : Nome della copia del backup.

Nel seguente esempio viene creata una copia di un backup automatico.

Per Linux, macOS o Unix:

```
aws elasticache copy-serverless-cache-snapshot \  
  --source-serverless-cache-snapshot-name automatic.my-cache-2023-11-27-03-15 \  
  --target-serverless-cache-snapshot-name automatic.my-cache-2023-11-27-03-15-copy
```

```
--target-serverless-cache-snapshot-name my-backup-copy
```

Per Windows:

```
aws elasticache copy-serverless-cache-snapshot ^  
  --source-serverless-cache-snapshot-name automatic.my-cache-2023-11-27-03-15 ^  
  --target-serverless-cache-snapshot-name my-backup-copy
```

Per ulteriori informazioni, consulta [copy-serverless-cache-snapshot](#) nella AWS CLI.

Copia di un backup del cluster progettato autonomamente (AWS CLI)

Per copiare il backup di un cluster progettato autonomamente utilizza l'operazione `copy-snapshot`.

Parametri

- `--source-snapshot-name` - Nome del backup da copiare.
- `--target-snapshot-name` - Nome della copia del backup.
- `--target-bucket` Riservato per l'esportazione di un backup. Non utilizzare questo parametro durante la creazione di una copia di un backup. Per ulteriori informazioni, consulta [Esportazione di un backup](#).

Nel seguente esempio viene creata una copia di un backup automatico.

Per Linux, macOS o Unix:

```
aws elasticache copy-snapshot \  
  --source-snapshot-name automatic.my-redis-primary-2014-03-27-03-15 \  
  --target-snapshot-name my-backup-copy
```

Per Windows:

```
aws elasticache copy-snapshot ^  
  --source-snapshot-name automatic.my-redis-primary-2014-03-27-03-15 ^  
  --target-snapshot-name my-backup-copy
```

Per ulteriori informazioni, consulta [copy-snapshot](#) nella AWS CLI.

Esportazione di un backup

Amazon ElastiCache supporta l'esportazione del ElastiCache backup in un bucket Amazon Simple Storage Service (Amazon S3), che consente di accedervi dall'esterno. ElastiCache Puoi esportare un backup utilizzando la ElastiCache console AWS CLI, l'o l'API. ElastiCache

L'esportazione di un backup può essere utile se è necessario avviare un cluster in un'altra AWS regione. 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 [Inizializzazione di un nuovo cluster progettato autonomamente con un backup creato esternamente](#). Un altro motivo per espandere i dati del cluster potrebbe essere per utilizzare il file .rdb per l'elaborazione offline.

Important

- Il ElastiCache backup e il bucket Amazon S3 in cui desideri copiarlo devono trovarsi nella stessa regione. AWS

Sebbene i backup copiati in un bucket Amazon S3 siano crittografati, ti consigliamo di non concedere altri accessi al bucket Amazon S3 in cui desideri archiviare i backup.

- L'esportazione di un backup in Amazon S3 non è supportata per i cluster che utilizzano il tiering di dati. Per ulteriori informazioni, consulta [Tiering di dati](#).

Prima di poter esportare un backup in un bucket Amazon S3, devi avere un bucket Amazon S3 nella stessa regione del backup. AWS Concedi l' ElastiCache accesso al bucket. Le prime due fasi mostrano come eseguire questa operazione.

Fase 1: creazione di un bucket Amazon S3

I passaggi seguenti utilizzano la console Amazon S3 per creare un bucket Amazon S3 in cui esportare e archiviare il backup. ElastiCache

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.

Il nome del bucket Amazon S3 deve essere conforme a DNS. Altrimenti, non ElastiCache potrai 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).
- b. Dall'elenco delle regioni, scegli una AWS regione per il tuo bucket Amazon S3. Questa AWS regione deve essere la stessa AWS del ElastiCache backup 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 ElastiCache l'accesso al bucket Amazon S3

ElastiCache Per poter copiare uno snapshot in un bucket Amazon S3, devi aggiornare la policy del bucket per ElastiCache concedere l'accesso al bucket.

Warning

Anche se i backup copiati in un bucket Amazon S3 sono crittografati, i dati sono accessibili a chiunque disponga dell'accesso al bucket Amazon S3. Pertanto, ti consigliamo di configurare le policy IAM per impedire l'accesso non autorizzato a questo bucket Amazon S3. Per ulteriori informazioni, consulta [Gestione accesso](#) nella Guida per gli utenti di Amazon S3.

Per creare le autorizzazioni corrette su un bucket Amazon S3, segui la procedura descritta di seguito.

Per concedere l'accesso a un bucket ElastiCache 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 desideri copiare il backup. Deve essere il bucket S3 che è stato creato in [Fase 1: creazione di un bucket Amazon S3](#).
3. Seleziona la scheda Autorizzazioni e sotto Autorizzazioni scegli Lista di controllo degli accessi (ACL), quindi seleziona Modifica.
4. Aggiungi l'ID canonico
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353
all'assegnatario con le opzioni seguenti:
 - Oggetti: Elenco, Scrittura
 - ACL bucket: Lettura, Scrittura

Note

- Per la GovCloud regione PDT, l'ID canonico è.
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
- Per la GovCloud regione OSU, l'ID canonico è.
c54286759d2a83da9c480405349819c993557275cf37d820d514b42da6893f5c

5. Selezionare Salva.

Passaggio 3: Esportazione e backup ElastiCache

Ora hai creato il tuo bucket S3 e concesso ElastiCache le autorizzazioni per accedervi. Successivamente, puoi utilizzare la ElastiCache console, la AWS CLI o l' ElastiCache API per esportare la tua istantanea al suo interno. Negli esempi seguenti, si presuppone che l'identità IAM del chiamante disponga delle seguenti autorizzazioni IAM aggiuntive, specifiche di S3.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketLocation",
      "s3:ListAllMyBuckets",
```

```

    "s3:PutObject",
    "s3:GetObject",
    "s3:DeleteObject",
    "s3:ListBucket"
  ],
  "Resource": "arn:aws:s3:::*"
}]
}

```

Per regioni attivate, l'esempio seguente simula l'aspetto di una policy aggiornata per il bucket S3. (Questo esempio utilizza la regione Asia Pacifico (Hong Kong).)

```

{
  "Version": "2012-10-17",
  "Id": "Policy15397346",
  "Statement": [
    {
      "Sid": "Stmt15399483",
      "Effect": "Allow",
      "Principal": {
        "Service": "elasticache.amazonaws.com"
      },
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::hkg-elasticache-backup",
        "arn:aws:s3:::hkg-elasticache-backup/*"
      ]
    },
    {
      "Sid": "Stmt15399484",
      "Effect": "Allow",
      "Principal": {
        "Service": "ap-east-1.elasticache-snapshot.amazonaws.com"
      },
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::hkg-elasticache-backup",
        "arn:aws:s3:::hkg-elasticache-backup/*"
      ]
    }
  ]
}

```

Esportazione di un ElastiCache backup (console)

I passaggi seguenti utilizzano la ElastiCache console per esportare un backup in un bucket Amazon S3 in modo da potervi accedere dall'esterno. ElastiCache Il bucket Amazon S3 deve trovarsi nella stessa AWS regione del backup. ElastiCache

Per esportare un ElastiCache backup in un bucket Amazon S3

1. [Accedi AWS Management Console e apri la ElastiCache console all'indirizzo https://console.aws.amazon.com/elasticache/.](https://console.aws.amazon.com/elasticache/)
2. Per visualizzare un elenco dei backup, dal riquadro di navigazione, scegliere Backups (Backup).
3. Dall'elenco di backup, scegli la casella a sinistra del nome del backup che desideri esportare.
4. Scegli Copia.
5. In Creare una copia del backup?, procedere come segue:
 - a. Nella casella Nome del nuovo backup, digita un nome per il nuovo backup.

Il nome deve essere compreso tra 1 e 1000 caratteri e dotato di codifica UTF-8.

ElastiCache aggiunge un identificatore di istanza e .rdb al valore che inserisci qui. Ad esempio, se si inserisce `my-exported-backup`, ElastiCache crea `my-exported-backup-0001.rdb`.

- b. Dall'elenco Posizione di destinazione S3, scegli il nome del bucket Amazon S3 in cui copiare il backup (il bucket creato in [Fase 1: creazione di un bucket Amazon S3](#)).

La posizione S3 di destinazione deve essere un bucket Amazon S3 nella regione AWS del backup con le seguenti autorizzazioni affinché il processo di esportazione abbia successo.

- Accesso agli oggetti : Lettura e Scrittura.
- Accesso alle autorizzazioni : Lettura.

Per ulteriori informazioni, consulta [Fase 2: concedere ElastiCache l'accesso al bucket Amazon S3](#).

- c. Scegli Copia.

Note

Se il tuo bucket S3 non dispone delle autorizzazioni necessarie per esportare un backup ElastiCache al suo interno, ricevi uno dei seguenti messaggi di errore. Tornare a [Fase 2: concedere ElastiCache l'accesso al bucket Amazon S3](#) per aggiungere le autorizzazioni specificate e riprovare a esportare il backup.

- ElastiCache non sono state concesse le autorizzazioni READ %s sul bucket S3.

Soluzione: aggiungere autorizzazioni di lettura sul bucket.

- ElastiCache non ha ricevuto le autorizzazioni di SCRITTURA %s sul bucket S3.

Soluzione: aggiungere autorizzazioni di scrittura sul bucket.

- ElastiCache non ha ricevuto le autorizzazioni READ_ACP %s sul bucket S3.

Soluzione: aggiungere Read (Lettura) per Accesso alle autorizzazioni sul bucket.

Se desideri copiare il backup in un'altra AWS regione, usa Amazon S3 per copiarlo. Per ulteriori informazioni, consulta [Copia un oggetto](#) nella Guida per l'utente di Amazon Simple Storage Service.

Esportazione di un backup ElastiCache senza server ()AWS CLI

Esportazione di un backup di una cache serverless

Esporta il backup in un bucket Amazon S3 utilizzando l'operazione CLI `export-serverless-cache-snapshot` con i seguenti parametri:

Parametri

- `--serverless-cache-snapshot-name` - Nome del backup da copiare.
- `--s3-bucket-name` : Nome del bucket Amazon S3 in cui desideri esportare il backup. Viene creata una copia del backup nel bucket specificato.
 - `--s3-bucket-name` Affinché il processo di esportazione abbia successo, deve essere un bucket Amazon S3 AWS nella regione del backup con le seguenti autorizzazioni.
 - Accesso agli oggetti : Lettura e Scrittura.
 - Accesso alle autorizzazioni : Lettura.

La seguente operazione copia un backup in my-s3-bucket.

Per Linux, macOS o Unix:

```
aws elasticache export-serverless-cache-snapshot \  
  --serverless-cache-snapshot-name automatic.my-redis-2023-11-27 \  
  --s3-bucket-name my-s3-bucket
```

Per Windows:

```
aws elasticache export-serverless-cache-snapshot ^  
  --serverless-cache-snapshot-name automatic.my-redis-2023-11-27 ^  
  --s3-bucket-name my-s3-bucket
```

Esportazione di un backup del cluster progettato autonomamente ElastiCache ()AWS CLI

Esportazione di un backup di un cluster progettato autonomamente

Esporta il backup in un bucket Amazon S3 utilizzando l'operazione CLI `copy-snapshot` con i seguenti parametri:

Parametri

- `--source-snapshot-name` - Nome del backup da copiare.
- `--target-snapshot-name` : Nome della copia del backup.

Il nome deve essere compreso tra 1 e 1000 caratteri e dotato di codifica UTF-8.

ElastiCache aggiunge un identificatore di istanza e `.rdb` al valore immesso qui. Ad esempio, se si inserisce `my-exported-backup`, ElastiCache crea `my-exported-backup-0001.rdb`.

- `--target-bucket` : Nome del bucket Amazon S3 in cui desideri esportare il backup. Viene creata una copia del backup nel bucket specificato.

`--target-bucket` Affinché il processo di esportazione abbia successo, deve essere un bucket Amazon S3 AWS nella regione del backup con le seguenti autorizzazioni.

- Accesso agli oggetti : Lettura e Scrittura.
- Accesso alle autorizzazioni : Lettura.

Per ulteriori informazioni, consulta [Fase 2: concedere ElastiCache l'accesso al bucket Amazon S3](#).

La seguente operazione copia un backup in my-s3-bucket.

Per Linux, macOS o Unix:

```
aws elasticache copy-snapshot \  
  --source-snapshot-name automatic.my-redis-primary-2016-06-27-03-15 \  
  --target-snapshot-name my-exported-backup \  
  --target-bucket my-s3-bucket
```

Per Windows:

```
aws elasticache copy-snapshot ^  
  --source-snapshot-name automatic.my-redis-primary-2016-06-27-03-15 ^  
  --target-snapshot-name my-exported-backup ^  
  --target-bucket my-s3-bucket
```

Ripristino da un backup in una nuova cache

È possibile ripristinare un backup esistente in una nuova cache serverless o in un cluster progettato autonomamente.

Ripristino di un backup in una cache serverless (console)

Note

ElastiCache serverless supporta i file RDB compatibili con le versioni di Redis comprese tra la 5.0 e l'ultima versione disponibile.

Per ripristinare un backup in una cache serverless (console)

1. Accedi alla AWS Management Console e apri la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione, scegliere Backups (Backup).
3. Nell'elenco dei backup, scegli la casella a sinistra del nome del backup di cui desideri eseguire il ripristino.
4. Scegli Operazioni e quindi Ripristina.
5. Immetti un nome e facoltativamente una descrizione per la nuova cache serverless.
6. Fai clic su Crea per creare la nuova cache e importare i dati dal backup.

Ripristino di un backup in un cluster progettato autonomamente (console)

Per ripristinare un backup in un cluster progettato autonomamente (console)

1. Accedi alla AWS Management Console e apri la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione, scegliere Backups (Backup).
3. Nell'elenco dei backup, scegli la casella a sinistra del nome di backup da cui desideri eseguire il ripristino.
4. Scegli Operazioni e quindi Ripristina.
5. Scegli Progetta la tua cache e personalizza le impostazioni del cluster, come il tipo di nodo, le dimensioni, il numero di partizioni, le repliche, il posizionamento delle zone di disponibilità e le impostazioni di sicurezza.

6. Scegli Crea per creare la nuova cache progettata autonomamente e importare i dati dal backup.

Ripristino di un backup in una cache serverless (AWS CLI)

Note

ElastiCache serverless supporta i file RDB compatibili con le versioni di Redis comprese tra la 5.0 e l'ultima versione disponibile.

Per ripristinare un backup in una nuova cache serverless (AWS CLI)

L'esempio della AWS CLI riportato di seguito crea una nuova cache utilizzando `create-serverless-cache` e importa i dati da un backup.

Per Linux, macOS o Unix:

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine redis \  
  --snapshot-arns-to-restore Snapshot-ARN
```

Per Windows:

```
aws elasticache create-serverless-cache ^ \  
  --serverless-cache-name CacheName ^ \  
  --engine redis ^ \  
  --snapshot-arns-to-restore Snapshot-ARN
```

Ripristino di un backup in un cluster progettato autonomamente (AWS CLI)

Per ripristinare un backup in un cluster progettato autonomamente (AWS CLI)

Puoi ripristinare un backup Redis (modalità cluster disabilitata) in due modi.

- Puoi ripristinare in un cluster Redis a nodo singolo (modalità cluster disabilitata) utilizzando l'operazione `create-cache-cluster` della AWS CLI.
- Puoi ripristinare in un cluster Redis con repliche di lettura (un gruppo di replica). Per svolgere questa operazione, puoi utilizzare Redis (modalità cluster disabilitata) o Redis (modalità cluster

abilitata) con l'operazione `create-replication-group` della AWS CLI. In questo caso, dai fonte al ripristino con un file Redis `.rdb`. Per ulteriori informazioni sull'inizializzazione di un nuovo cluster progettato autonomamente, consulta [Inizializzazione di un nuovo cluster progettato autonomamente con un backup creato esternamente](#).

Quando utilizzi l'operazione `create-cache-cluster` o `create-replication-group`, assicurati di includere il parametro `--snapshot-name` o `--snapshot-arn` per inizializzare il nuovo cluster o gruppo di replica con i dati del backup.

Eliminazione di un backup

Un backup automatico viene automaticamente eliminato quando il suo periodo di retention scade. Se elimini un cluster, vengono anche eliminati tutti i relativi backup automatici. Se elimini un gruppo di replica, vengono anche eliminati tutti i backup automatici dai cluster in tale gruppo.

ElastiCache fornisce un'operazione API di eliminazione che consente di eliminare un backup in qualsiasi momento, a prescindere che il backup sia stato creato automaticamente o manualmente. Poiché i backup manuali non dispongono di un periodo di conservazione, l'eliminazione manuale è il solo modo per rimuoverli.

Puoi eliminare un backup utilizzando la console ElastiCache, il AWS CLI o l'API ElastiCache.

Eliminazione di un backup (console)

La procedura seguente consente di eliminare un backup utilizzando la console ElastiCache.

Per eliminare un backup

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegliere Backups (Backup).

Viene visualizzata la schermata dei backup con un elenco dei backup.
3. Scegliere la casella a sinistra del nome del backup da eliminare.
4. Scegliere Delete (Elimina).
5. Per eliminare questo backup, scegliere Delete (Elimina) nella schermata di conferma Delete Backup (Elimina backup). Lo stato cambia in eliminazione.

Eliminazione di un backup serverless (AWS CLI)

Utilizza l'operazione `delete-snapshot` della AWS CLI con il seguente parametro per eliminare un backup serverless.

- `--serverless-cache-snapshot-name` - Nome del backup da eliminare.

Il codice seguente elimina il backup `myBackup`.

```
aws elasticache delete-serverless-cache-snapshot --serverless-cache-snapshot-name myBackup
```

Per ulteriori informazioni, consulta [delete-serverless-cache-snapshot](#) nella Guida di riferimento dei comandi di AWS CLI.

Eliminazione di un backup del cluster progettato autonomamente (AWS CLI)

Utilizza l'operazione `delete-snapshot` della AWS CLI con il seguente parametro per eliminare un backup del cluster progettato autonomamente.

- `--snapshot-name` - Nome del backup da eliminare.

Il codice seguente elimina il backup `myBackup`.

```
aws elasticache delete-snapshot --snapshot-name myBackup
```

Per ulteriori informazioni, vedere [delete-snapshot](#) in AWS CLI guida di riferimento del comando.

Assegnazione di tag di backup

Puoi assegnare metadati a ciascun backup sotto forma di tag. I tag consentono di categorizzare le tue risorse di backup 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. Per ulteriori informazioni, consulta [Risorse cui è possibile associare tag](#).

I tag per l'allocazione dei costi sono un mezzo di tracciamento dei costi tra più servizi AWS raggruppando le spese su 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 ElastiCache, la AWS CLI o l'API ElastiCache puoi aggiungere, elencare, modificare, rimuovere o copiare tag di allocazione dei costi sui backup. Per ulteriori informazioni, consulta [Monitoraggio dei costi con i tag di allocazione dei costi](#).

Inizializzazione di un nuovo cluster progettato autonomamente con un backup creato esternamente

Quando crei un nuovo cluster progettato autonomamente Redis, puoi iniziarlo con i dati di un file di backup .rdb Redis. L'inizializzazione del cluster è utile se gestisci attualmente un'istanza Redis esterna a ElastiCache e desideri compilare il nuovo cluster progettato autonomamente ElastiCache per Redis con i dati Redis esistenti.

Per inizializzare un nuovo cluster progettato autonomamente Redis da un backup Redis creato all'interno di Amazon ElastiCache, consulta [Ripristino da un backup in una nuova cache](#).

Quando utilizzi un file .rdb Redis per inizializzare un cluster progettato autonomamente Redis, puoi procedere come segue:

- Esegui l'aggiornamento da un cluster non partizionato in un cluster progettato autonomamente Redis (modalità cluster abilitata) che esegue Redis versione 3.2.4.
- Specifica un numero di partizioni (dette gruppi di nodi nell'API e nella CLI) nel nuovo cluster progettato autonomamente. Questo numero può essere diverso dal numero di partizioni nel cluster progettato autonomamente che era stato utilizzato per creare il file di backup.
- Specifica un tipo di nodo diverso per il nuovo cluster progettato autonomamente, maggiore o minore rispetto a quello utilizzato nel cluster che ha eseguito il backup. 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. Per ulteriori informazioni, consulta [Verifica della disponibilità di memoria sufficiente per creare uno snapshot Redis](#).
- Distribuisci le chiavi negli slot del nuovo cluster Redis (modalità cluster abilitata) in maniera diversa rispetto al cluster che è stato utilizzato per creare il file di backup.

Note

Non puoi inizializzare un cluster Redis (modalità cluster disabilitata) da un file .rdb creato da un cluster Redis (modalità cluster abilitata).

Important

- Assicurati che i dati di backup Redis non superino le risorse del nodo. Ad esempio, non puoi caricare un file .rdb con 5 GB di dati Redis in un nodo cache.m3.medium che dispone di 2,9 GB di memoria.

Se il backup è troppo grande, lo stato del cluster risultante sarà `restore-failed`. In tal caso, occorre eliminare il cluster e ricominciare.

Per un elenco completo di tipi di nodi e specifiche, consulta [Parametri specifici del tipo di nodo di Redis](#) e [caratteristiche e dettagli prodotto Amazon ElastiCache](#).

- 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](#).

Di seguito sono riportati gli argomenti che illustrano in maniera dettagliata la migrazione del cluster Redis dall'esterno di ElastiCache for Redis a ElastiCache for Redis.

Migrazione in ElastiCache for Redis

- [Fase 1: creazione di un backup Redis](#)
- [Fase 2: creazione di un bucket Amazon S3 e una cartella](#)
- [Fase 3: caricamento del backup in Amazon S3](#)
- [Fase 4: concessione a ElastiCache dell'accesso in lettura al file .rdb](#)

Fase 1: creazione di un backup Redis

Per creare un backup Redis per dare fonte all'istanza ElastiCache for Redis

1. Connettersi all'istanza Redis esistente.
2. Eseguire l'operazione BGSAVE or SAVE Redis per creare un backup. 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 backup, consulta [Redis persistence](#) presso il sito Web Redis.

Fase 2: creazione di un bucket Amazon S3 e una cartella

Dopo che è stato creato, il file di backup deve essere caricato 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 [Fase 3: caricamento del backup in Amazon S3](#).


Come creare un bucket Amazon S3.

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo <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, ElastiCache non può 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).

Devi creare il bucket Amazon S3 nella stessa regione AWS come nuovo cluster ElastiCache for Redis. Questo approccio assicura la massima velocità di trasferimento dei dati quando ElastiCache 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 ancora consentire al bucket e i suoi contenuti devono essere utilizzati per dare fonte al nuovo cluster Redis.

Per aggiungere una cartella a un bucket Amazon S3

1. Accedi alla AWS Management Console e apri la console di Amazon S3 all'indirizzo <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. Seleziona Salva.

Prendi nota del nome del bucket e del nome della cartella.

Fase 3: caricamento del backup in Amazon S3

Ora, carica il file .rdb creato in [Fase 1: creazione di un backup Redis](#). 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 questo processo, consulta [Aggiungere un oggetto in un bucket](#). Tra le fasi 2 e 3, scegliere il nome della cartella creata.

Per caricare il file .rdb in una cartella Amazon S3

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo <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. Verificare che il file o i file corretti sono elencati nella finestra di dialogo Upload (Carica), quindi scegliere Upload (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 richiesto per inizializzare il nuovo cluster con i dati in questo backup.

Per ulteriori informazioni, consulta [Restrizioni e limitazioni dei bucket](#) nella Guida per gli utenti di Amazon Simple Storage Service.

Fase 4: concessione a ElastiCache dell'accesso in lettura al file .rdb

Ora, concedere a ElastiCache l'accesso in lettura al file di backup .rdb. L'accesso di ElastiCache al file di backup viene concesso in modo diverso a seconda se il bucket si trova in una regione AWS di default o in una regione AWS scelta.

Le regioni AWS introdotte prima del 20 marzo 2019 sono abilitate per impostazione predefinita. È possibile iniziare a lavorare in queste regioni AWS immediatamente. Le regioni introdotte dopo il 20 marzo 2019, come l'Asia Pacifico (Hong Kong) e Medio Oriente (Bahrein), sono disabilitate per impostazione predefinita. È necessario abilitare o scegliere tali regioni prima di poterle utilizzare, come descritto in [Managing AWS regions](#) (Gestione delle regioni AWS) nella Riferimenti generali di AWS.

Scegliere l'approccio in base alla propria regione AWS:

- Per una regione predefinita, utilizzare la procedura in [Concedere a ElastiCache l'accesso in lettura al file.rdb in una regione predefinita](#).
- Per una regione scelta, utilizzare la procedura in [Concedere l'accesso in lettura ElastiCache al file.rdb in una regione scelta](#).

Concedere a ElastiCache l'accesso in lettura al file.rdb in una regione predefinita


Le regioni AWS introdotte prima del 20 marzo 2019 sono abilitate per impostazione predefinita. È possibile iniziare a lavorare in queste regioni AWS immediatamente. Le regioni introdotte dopo il 20 marzo 2019, come l'Asia Pacifico (Hong Kong) e Medio Oriente (Bahrein), sono disabilitate per impostazione predefinita. È necessario abilitare o scegliere tali regioni prima di poterle utilizzare,

come descritto in [Managing AWS regions](#) (Gestione delle regioni AWS) nella Riferimenti generali di AWS.

Concedere a ElastiCache l'accesso in lettura al file di backup in una regione AWS abilitata per impostazione predefinita

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo <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. Scegliere il nome del file di backup .rdb. Il nome del file selezionato viene visualizzato sopra le schede nella parte superiore della pagina.
5. Seleziona Autorizzazioni.
6. Se `aws-scs-s3-readonly` o uno degli ID canonici nell'elenco seguente non è elencato come un utente, procedere come segue:
 - a. In Accesso per altri account AWS, scegliere `Add grantees` (Aggiungi beneficiario).
 - b. Nella casella, aggiungere l'ID canonico della regione AWS come mostrato nel seguente elenco:
 - AWS Regione GovCloud (Stati Uniti occidentali):

```
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
```

 Important

Il backup deve trovarsi in un bucket S3 in AWS GovCloud (US) per poterlo scaricare in un cluster Redis in AWS GovCloud (US).

- Regioni AWS abilitate per impostazione predefinita:

```
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353
```

- c. Impostare le autorizzazioni sul bucket scegliendo `Yes (Sì)` per:
 - Oggetto elenco/scrittura
 - Autorizzazioni ACL oggetto lettura/scrittura

- d. Seleziona Salva.
7. Scegliere Overview (Panoramica), quindi Download (Scarica).

Concedere l'accesso in lettura ElastiCache al file.rdb in una regione scelta

Le regioni AWS introdotte prima del 20 marzo 2019 sono abilitate per impostazione predefinita. È possibile iniziare a lavorare in queste regioni AWS immediatamente. Le regioni introdotte dopo il 20 marzo 2019, come l'Asia Pacifico (Hong Kong) e Medio Oriente (Bahrein), sono disabilitate per impostazione predefinita. È necessario abilitare o scegliere tali regioni prima di poterle utilizzare, come descritto in [Managing AWS regions](#) (Gestione delle regioni AWS) nella Riferimenti generali di AWS.

Ora, concedere a ElastiCache l'accesso in lettura al file di backup .rdb.

Per concedere l'accesso in lettura ElastiCache al file di backup

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo <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. Scegliere il nome del file di backup .rdb. Il nome del file selezionato viene visualizzato sopra le schede nella parte superiore della pagina.
5. Scegli la scheda Permissions (Autorizzazioni).
6. Sotto Autorizzazioni, scegli Policy bucket e seleziona Modifica.
7. Aggiornare la policy per concedere a ElastiCache le autorizzazioni necessarie per eseguire le operazioni:
 - Aggiungere ["Service" : "*region-full-name*.elasticache-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": "ap-east-1.elasticache-snapshot.amazonaws.com"
      },
      "Action": [
        "s3:GetObject",
        "s3:ListBucket",
        "s3:GetBucketAcl"
      ],
      "Resource": [
        "arn:aws:s3:::example-bucket",
        "arn:aws:s3:::example-bucket/backup1.rdb",
        "arn:aws:s3:::example-bucket/backup2.rdb"
      ]
    }
  ]
}
```

8. Seleziona Salva modifiche.

Fase 5: inizializzazione del cluster ElastiCache con i dati del file .rdb

Puoi ora creare un cluster ElastiCache e inizializzarlo con i dati del file .rdb. Per creare il cluster, segui le istruzioni in [Creazione di un cluster](#) o [Creazione di un gruppo di replica Redis da zero](#). Assicurati di scegliere Redis come motore del cluster.

Il metodo utilizzato per indicare a ElastiCache dove trovare il backup Redis caricato in Amazon S3 dipende dal metodo che utilizzi per creare il cluster:

Inizializzazione del cluster ElastiCache o gruppo di replica con i dati del file .rdb

- Utilizzo della console ElastiCache

Quando selezioni Cluster settings (Impostazioni cluster), scegli Restore from backups (Ripristina dai backup) come metodo di creazione del cluster, quindi scegli Other backups (Altri backup) come Source (Origine) nella sezione Backup source (Fonte del backup). 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*.

- Utilizzando la AWS CLI

Se utilizzi l'operazione `create-cache-cluster` o `create-replication-group`, 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 backup archiviati in Amazon S3.

- Utilizzo dell'API ElastiCache

Se utilizzi l'operazione `CreateCacheCluster` o API ElastiCache `CreateReplicationGroup`, utilizza 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 backup archiviati in Amazon S3.

Important

Durante l'inizializzazione di un cluster Redis (modalità cluster abilitata), occorre configurare ogni gruppo di nodi (partizione) nel nuovo cluster o gruppo di replica. A questo proposito, utilizza il parametro `--node-group-configuration` (API: `NodeGroupConfiguration`). Per ulteriori informazioni, consulta gli argomenti seguenti:

- CLI: [create-replication-group](#) nella documentazione di riferimento AWS CLI
- API: [CreateReplicationGroup](#) nella documentazione di riferimento dell'API ElastiCache

Durante il processo di creazione del cluster, i dati nel backup Redis vengono scritti nel cluster. Puoi monitorare l'avanzamento visualizzando i messaggi di evento ElastiCache. A questo scopo, consulta la console ElastiCache e scegliere Cache Events (Eventi cache). Puoi anche utilizzare l'interfaccia a riga di comando ElastiCache AWS o l'API ElastiCache per ottenere messaggi evento. Per ulteriori informazioni, consulta [Visualizzazione di eventi ElastiCache C](#).

Versioni e aggiornamenti del motore

Questa sezione illustra e spiega come aggiornare le versioni del motore Redis supportate.

Argomenti

- [Versioni e aggiornamenti del motore](#)
- [ElastiCache supportate per le versioni Redis](#)
- [Pianificazione della fine del ciclo di vita delle versioni di Redis](#)
- [Come aggiornare la versione di un motore](#)
- [Risolvere il blocco di un aggiornamento del motore Redis](#)
- [Differenze di compatibilità e comportamento delle versioni principali](#)

Versioni e aggiornamenti del motore

Le versioni di ElastiCache per Redis sono identificate con una versione semantica che comprende la componente PRINCIPALE e SECONDARIA. Ad esempio, in Redis 6.2 la versione principale è 6 e la versione secondaria 2. Quando si utilizzano cluster progettati autonomamente, ElastiCache per Redis espone anche la componente PATCH, ad esempio in Redis 6.2.1 la versione della patch è 1.

Le versioni MAJOR sono per modifiche incompatibili con l'API e le versioni MINOR sono per nuove funzionalità aggiunte per compatibilità a ritroso. Le versioni PATCH sono per correzioni di bug compatibili a ritroso e modifiche non funzionali.

Gestione delle versioni per ElastiCache serverless

ElastiCache serverless applica automaticamente alla cache l'ultima versione SECONDARIA e della PATCH del software, senza alcun impatto o tempi di inattività dell'applicazione. Non è richiesta nessuna azione da parte tua.

Quando è disponibile una nuova versione PRINCIPALE, ElastiCache serverless invia una notifica nella console e un evento in EventBridge. Puoi scegliere di aggiornare la cache all'ultima versione principale utilizzando la console, la CLI o l'API e selezionando la versione più recente del motore.

Gestione delle versioni per cluster ElastiCache progettati autonomamente

Quando si usano cluster ElastiCache progettati autonomamente, è possibile controllare quando aggiornare il software di un cluster di cache a una nuova versione supportata da ElastiCache. Puoi controllare quando aggiornare la cache all'ultima versione PRINCIPALE, SECONDARIA e della PATCH disponibili. L'utente può eseguire l'aggiornamento a una versione del motore sul cluster o gruppo di replica modificando quest'ultimo e specificando la nuova versione da utilizzare.

Puoi controllare se e quando aggiornare il software conforme al protocollo di un cluster di cache a una nuova versione supportata da ElastiCache. 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.

Poiché presentano rischi relativi alla compatibilità, gli aggiornamenti delle versioni non vengono eseguiti automaticamente, ma devono essere avviati manualmente.

L'utente può eseguire l'aggiornamento a una versione del motore sul cluster o gruppo di replica modificando quest'ultimo e specificando la nuova versione da utilizzare. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Modifica dei cluster](#)
- [Modifica di un gruppo di replica](#)

Considerazioni sull'aggiornamento quando si usano cluster progettati autonomamente

Note

Le seguenti considerazioni si applicano solo quando si aggiornano cluster progettati autonomamente. Non si applicano a ElastiCache serverless.

Quando aggiorni un cluster progettato autonomamente, considera quanto indicato di seguito.

- La gestione della versione del motore è progettata in modo da avere il maggior controllo possibile sulle modalità di applicazione delle patch. Tuttavia, ElastiCache si riserva il diritto di applicare patch al cluster per tuo conto in caso di vulnerabilità critiche della sicurezza nel software del sistema e della cache.
- A partire da Redis 6.0, ElastiCache per Redis offrirà una versione singola per ogni release principale Redis OSS, piuttosto che più versioni secondarie.
- A partire dalla versione 5.0.6 del motore 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.
- È inoltre possibile aggiornare i cluster ElastiCache con versioni precedenti alla 5.0.6. Il processo utilizzato è lo stesso, ma potrebbe richiedere tempi di failover più lunghi durante la propagazione DNS (da 30 secondi a un minuto).
- A partire da Redis 7, ElastiCache per Redis supporta il passaggio tra Redis (modalità cluster disabilitata) e Redis (modalità cluster abilitata).
- Il processo di aggiornamento del motore Amazon ElastiCache for Redis è progettato per cercare di mantenere i dati esistenti e, a tale scopo, richiede una corretta replica Redis.
- Quando si aggiorna il motore, ElastiCache per Redis interromperà le connessioni client esistenti. Per ridurre al minimo i tempi di inattività durante gli aggiornamenti del motore, si consiglia di implementare [le best practice per i client Redis](#) con ripetizioni dei tentativi in caso di errore e backoff esponenziale e le best practice per [ridurre al minimo i tempi di inattività durante la manutenzione](#).

- Non puoi passare direttamente da Redis (modalità cluster disabilitata) a Redis (modalità cluster abilitata) quando esegui l'aggiornamento del motore. La procedura seguente mostra come eseguire l'aggiornamento da Redis (modalità cluster disabilitata) a Redis (modalità cluster abilitata).

Per eseguire l'aggiornamento da una versione del motore Redis (modalità cluster disabilitata) a Redis (modalità cluster abilitata)

1. Effettuare il backup del gruppo di replica o cluster Redis (modalità cluster disabilitata). Per ulteriori informazioni, consulta [Esecuzione di backup manuali](#).
 2. Avvalendosi del backup, creare e configurare un cluster Redis (modalità cluster abilitata) con una partizione (gruppo di nodi). Specificare la nuova versione del motore e abilitare la modalità cluster durante la creazione del cluster o gruppo di replica. Per ulteriori informazioni, consulta [Inizializzazione di un nuovo cluster progettato autonomamente con un backup creato esternamente](#).
 3. Eliminare il gruppo di replica o cluster Redis (modalità cluster disabilitata). Per ulteriori informazioni, consulta [Eliminazione di un cluster](#) o [Eliminazione di un gruppo di replica](#).
 4. Dimensionare il nuovo gruppo di replica o cluster Redis (modalità cluster abilitata) in base al numero delle partizioni (gruppi di nodi) necessari. Per ulteriori informazioni, consulta [Ridimensionamento dei cluster in Redis \(modalità cluster attivata\)](#)
- Quando si aggiornano le versioni principali del motore, ad esempio da 5.0.6 a 6.0, è necessario scegliere anche un nuovo gruppo di parametri compatibile con la nuova versione del motore.
 - Per i cluster a nodo singolo di Redis nonché i cluster con la funzione Multi-AZ disabilitata, consigliamo di programmare una quantità di memoria libera sufficiente, secondo quanto descritto in [Verifica della disponibilità di memoria sufficiente per creare uno snapshot Redis](#). In condizioni simili, il nodo primario non sarà a disposizione delle richieste di servizi durante la procedura di aggiornamento.
 - Per i cluster Redis con la funzione Multi-AZ abilitata, consigliamo inoltre di pianificare aggiornamenti del motore nei periodi con scarso traffico di scrittura in entrata. Durante l'aggiornamento a Redis 5.0.6 o versioni successive, il cluster primario continua a essere disponibile per servire richieste durante il processo di aggiornamento.

I cluster e gruppi di replica con più partizioni vengono elaborati e dotati di patch come di seguito:

- Tutti le partizioni vengono elaborati in parallelo. Ognle partizioni ammette un'unica operazione di aggiornamento 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.
- Se sul cluster o gruppo di replica in uso sono abilitate le crittografie, non è possibile eseguire l'aggiornamento a una versione del motore che non le supporti come ad esempio, da 3.2.6 a 3.2.10.

Come aggiornare la versione di un motore

L'utente può eseguire l'aggiornamento a una versione sul cluster o gruppo di replica, modificando quest'ultimo con la console ElastiCache, lo AWS CLI o l'API di ElastiCache e specificando la nuova versione del motore da utilizzare. Per ulteriori informazioni, consulta i seguenti argomenti.

Come modificare cluster e gruppi di replica	
Cluster	Gruppi di replica
Utilizzando il AWS Management Console	Utilizzo di AWS Management Console
Utilizzando il AWS CLI	Utilizzo di AWS CLI
Utilizzo dell' ElastiCache API	Utilizzo dell'API ElastiCache

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

Operazioni in sospeso	Operazioni bloccate
	Aggiornamento immediato del motore

Per risolvere il blocco di un aggiornamento del motore Redis

- Esegui una di queste operazioni:
 - Programmare l'operazione di aggiornamento del motore Redis in una finestra di manutenzione successiva, deselegionando la casella di controllo Apply immediately (Applica immediatamente).

Con la CLI utilizzare `--no-apply-immediately`. Con l'API, utilizzare `ApplyImmediately=false`.

- Attendere una finestra di manutenzione successiva per l'esecuzione dell'aggiornamento del motore Redis.
- Aggiungere l'operazione di dimensionamento Redis alle modifiche del cluster con la casella di controllo Apply Immediately (Applica immediatamente) selezionata.

Con la CLI utilizzare `--apply-immediately`. Con l'API, utilizzare `ApplyImmediately=true`.

Questo approccio annulla efficacemente l'aggiornamento del motore nella finestra di manutenzione successiva, poiché lo si esegue immediatamente.

ElastiCache supportate per le versioni Redis

Le cache ElastiCache serverless supportano le seguenti versioni Redis:

- [ElastiCache per Redis versione 7.1 \(avanzata\)](#)

I cluster ElastiCache progettati autonomamente supportano le seguenti versioni Redis:

- [ElastiCache per Redis versione 7.1 \(avanzata\)](#)
- [ElastiCache per Redis versione 7.0 \(avanzata\)](#)
- [ElastiCache for Redis versione 6.2 \(avanzato\)](#)
- [ElastiCache for Redis versione 6.0 \(avanzata\)](#)
- [ElastiCache for Redis versione 5.0.6 \(potenziata\)](#)
- [ElastiCache per Redis versione 5.0.5 \(obsoleta, usa versione 5.0.6\)](#)
- [ElastiCache per Redis versione 5.0.4 \(obsoleta, usa versione 5.0.6\)](#)
- [ElastiCache per Redis versione 5.0.3 \(obsoleta, usa versione 5.0.6\)](#)
- [ElastiCache per Redis versione 5.0.0 \(obsoleta, usa versione 5.0.6\)](#)
- [ElastiCache for Redis versione 4.0.10 \(potenziata\)](#)
- [Versioni precedenti di fine del ciclo di vita \(EOL\) \(3.x\)](#)
- [Versioni precedenti di fine del ciclo di vita \(EOL\) \(2.x\)](#)

ElastiCache per Redis versione 7.1 (avanzata)

Questa versione include miglioramenti delle prestazioni che consentono ai carichi di lavoro di aumentare la velocità di trasmissione effettiva e ridurre le latenze operative. ElastiCache 7.1 introduce [due miglioramenti principali](#):

Abbiamo esteso la funzionalità dei thread di I/O avanzato per gestire anche la logica del livello di presentazione. Per livello di presentazione, intendiamo i thread di I/O avanzato che non solo leggono l'input del client, ma analizzano anche l'input nel formato di comando binario Redis. L'input viene quindi inoltrato al thread principale per l'esecuzione, fornendo così un aumento delle prestazioni. Pattern di accesso alla memoria Redis migliorato. Le fasi di esecuzione di molte operazioni sulla struttura dei dati sono interleaved, per garantire l'accesso parallelo alla memoria e una latenza di accesso alla memoria ridotta. Quando eseguono ElastiCache su nodi R7g.4xLarge basati su Graviton3 o più grandi, i clienti possono raggiungere oltre 1 milione di richieste al secondo per nodo. Grazie ai miglioramenti delle prestazioni di ElastiCache per Redis v7.1, i clienti possono ottenere

fino al 100% in più di velocità di trasmissione effettiva e una latenza P99 inferiore del 50% rispetto a ElastiCache per Redis v7.0. Questi miglioramenti sono abilitati su nodi di dimensioni con almeno 8 core fisici (2x1large su Graviton e 4x1large su x86), indipendentemente dal tipo di CPU e non richiedono modifiche al client.

Note

ElastiCache v7.1 è compatibile con OSS Redis v7.0.

ElastiCache per Redis versione 7.0 (avanzata)

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

- [Funzioni Redis](#): ElastiCache per Redis 7 aggiunge il supporto per Funzioni Redis e fornisce un'esperienza gestita che consente agli sviluppatori di eseguire [script LUA](#) con la logica dell'applicazione archiviata nel cluster ElastiCache, senza richiedere ai client di inviare nuovamente gli script al server con ogni Connessione.
- [Miglioramenti ACL](#): ElastiCache per Redis 7 aggiunge il supporto per la prossima versione della lista di controllo degli accessi (ACL) Redis. Con ElastiCache per Redis 7, i client possono ora specificare più set di autorizzazioni su chiavi o keypace specifici in Redis.
- [Pub/Sub partizionato](#): ElastiCache per Redis 7 aggiunge il supporto per eseguire la funzionalità Redis Pub/Sub in modo partizionato quando si esegue ElastiCache in Modalità cluster abilitata (CME). Le funzionalità Redis Pub/Sub consentono agli editori di inviare messaggi a qualsiasi numero di sottoscrittori su un canale. Con Amazon ElastiCache per Redis 7, i canali sono associati a una partizione nel cluster ElastiCache. Questo elimina la necessità di propagare le informazioni sui canali tra le partizioni con conseguente miglioramento della scalabilità.
- [Multiplexing I/O avanzato](#): ElastiCache per Redis versione 7 introduce il multiplexing I/O avanzato, che offre una maggiore velocità di trasmissione effettiva e una latenza ridotta per carichi di lavoro ad alta velocità di trasmissione effettiva con molte connessioni client simultanee a un cluster ElastiCache. Ad esempio, quando si utilizza un cluster di nodi r6g.xlarge e si eseguono 5200 client simultanei, è possibile ottenere fino al 72% di aumento della velocità di trasmissione effettiva (operazioni di lettura e scrittura al secondo) e una riduzione della latenza P99 fino al 71%, rispetto a ElastiCache per Redis versione 6.

Per ulteriori informazioni sul rilascio Redis 7.0, consulta le [Note di rilascio di Redis 7.0](#) alla pagina Redis di GitHub.

ElastiCache for Redis versione 6.2 (avanzato)

ElastiCache for Redis 6.2 include miglioramenti delle prestazioni per cluster abilitati TLS che utilizzano tipi di nodi x86 con 8 o più vCPU o tipi di nodi Graviton2 con 4 o più vCPU. Questi miglioramenti migliorano la velocità effettiva e riducono i tempi necessari per stabilire la connessione client scaricando la crittografia su altre vCPU. Con Redis 6.2, è possibile anche gestire l'accesso ai canali Pub/Sub con le regole della Lista di controllo accessi (ACL).

Con questa versione, introduciamo inoltre il supporto per il tiering di dati su nodi cluster contenenti lo SSD NVMe collegato in locale. Per ulteriori informazioni, consulta [Tiering di dati](#).

La versione 6.2.6 del motore Redis introduce anche il supporto per il formato JSON (JavaScript Object Notation) nativo, un modo semplice e senza schemi per codificare set di dati complessi nei 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 su JSON](#). Sono incluse anche le metriche relative a JSON, `JsonBasedCmds` e `JsonBasedCmdsLatency`, incorporate in CloudWatch per monitorare l'utilizzo di questo tipo di dati. Per ulteriori informazioni, consulta [Parametri di Redis](#).

Utilizzando 6.2 è possibile specificare la versione del motore. ElastiCache for Redis richiama automaticamente la versione secondaria preferita di Redis 6.2 disponibile. Ad esempio, quando si crea/modifica un cluster di cache, si imposta il parametro `--engine-version` su 6.2. Il cluster verrà avviato con l'attuale versione secondaria preferita di Redis 6.2 al momento della creazione/modifica. Se si specifica la versione 6.x del motore nell'API, l'ultima versione secondaria di Redis 6.

Per i cluster 6.0 esistenti, è possibile accedere al successivo aggiornamento automatico della versione secondaria impostando il parametro `AutoMinorVersionUpgrade` su `yes` nell'API `CreateCacheCluster`, `ModifyCacheCluster`, `CreateReplicationGroup` o `ModifyReplicationGroup`. ElastiCache for Redis aggiornerà la versione secondaria dei cluster 6.0 esistenti a 6.2 utilizzando gli aggiornamenti self-service. Per ulteriori informazioni, consulta [Aggiornamenti self-service in Amazon ElastiCache](#).

Quando si chiama l'API `DescribeCacheEngineVersions`, il valore del parametro `EngineVersion` verrà impostato su 6.2 e la versione effettiva del motore con la versione secondaria verrà restituita nel campo `CacheEngineVersionDescription`.

Per ulteriori informazioni sulle note di rilascio di Redis 6.2, consulta le [Note di rilascio di Redis 6.2](#) alla pagina Redis di GitHub.

ElastiCache for Redis versione 6.0 (avanzata)

Amazon ElastiCache for Redis presenta la nuova versione del motore Redis, che include [l'Autenticazione degli utenti con il controllo degli accessi basato sui ruoli](#), il caching lato client e significativi miglioramenti operativi.

A partire da Redis 6.0, ElastiCache for Redis offrirà una singola versione per ogni release principale Redis OSS, piuttosto che offrire più versioni secondarie. ElastiCache for Redis gestirà automaticamente la versione secondaria dei cluster di cache in esecuzione, garantendo prestazioni migliorate e maggiore sicurezza.

È inoltre possibile accedere al prossimo aggiornamento automatico della versione secondaria impostando il parametro `AutoMinorVersionUpgrade` su `yes` ed ElastiCache for Redis gestirà l'aggiornamento della versione secondaria, tramite aggiornamenti self-service. Per ulteriori informazioni, consulta [Aggiornamenti del servizio in ElastiCache](#).

È possibile specificare la versione del motore utilizzando `6.0`. ElastiCache for Redis richiama automaticamente la versione secondaria preferita di Redis 6.0 disponibile. Ad esempio, quando si crea/modifica un cluster di cache, si imposta il parametro `--engine-version` su `6.0`. Il cluster verrà avviato con l'attuale versione minore preferita di Redis 6.0 al momento della creazione/modifica. Qualsiasi richiesta con un valore di versione secondaria specifico verrà rifiutata, verrà generata un'eccezione e il processo avrà esito negativo.

Quando si chiama l'API `DescribeCacheEngineVersions`, il valore del parametro `EngineVersion` verrà impostato su `6.0` e la versione effettiva del motore con la versione secondaria verrà restituita nel campo `CacheEngineVersionDescription`.

Per ulteriori informazioni sul rilascio Redis 6.0, consulta le [Note di rilascio di Redis 6.0](#) alla pagina Redis di GitHub.

ElastiCache for Redis versione 5.0.6 (potenziata)

Amazon ElastiCache per Redis presenta la nuova versione del motore Redis, che include correzioni di bug e i seguenti aggiornamenti cumulativi:

- Garanzia di stabilità del motore in condizioni speciali.
- Miglioramento della gestione degli errori di Hyperloglog.
- Comandi avanzati di handshake per una replica affidabile.
- Tracciamento costante della consegna dei messaggi tramite il comando `XCLAIM`.

- Gestione migliorata del campo LFU negli oggetti.
- Gestione migliorata delle transazioni durante l'utilizzo ZPOP.
- Possibilità di rinominare comandi: un parametro denominato `rename-commands` che consente di rinominare i comandi Redis potenzialmente pericolosi o costosi che possono causare la perdita accidentale dei dati, ad esempio `FLUSHALL` o `FLUSHDB`. È simile alla configurazione di `rename-command` della versione open source di Redis. Tuttavia, ElastiCache ha migliorato l'esperienza fornendo un flusso di lavoro completamente gestito. Le modifiche al nome del comando vengono applicate immediatamente e propagate automaticamente su tutti i nodi del cluster che contengono l'elenco comandi. Non è richiesto alcun intervento da parte dell'utente, come il riavvio dei nodi.

I seguenti esempi mostrano come modificare i gruppi di parametri esistenti. È incluso il parametro `rename-commands` che è un elenco di comandi separati da spazi da rinominare:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall restrictedflushall'" --region region
```

In questo esempio, il parametro `rename-commands` viene utilizzato per rinominare il comando `flushall` in `restrictedflushall`.

Per rinominare più comandi, usa:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall restrictedflushall flushdb restrictedflushdb'" --region region
```

Per ripristinare una modifica, esegui nuovamente il comando ed escludi i valori rinominati dall'elenco `ParameterValue` che vuoi conservare, come illustrato di seguito:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall restrictedflushall'" --region region
```

In questo caso, il comando `flushall` viene rinominato in `restrictedflushall` e qualsiasi altro comando rinominato riacquista il nome originale.

Note

Quando si rinominano i comandi, si applicano le seguenti limitazioni:

- Tutti i comandi rinominati devono essere alfanumerici.
- La lunghezza massima dei nuovi nomi di comandi è di 20 caratteri alfanumerici.
- Durante la ridenominazione dei comandi, assicurati di aggiornare il gruppo di parametri associato al cluster.
- Per impedire completamente l'uso di un comando, usa la parola chiave `blocked`, come indicato di seguito:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall blocked'" --region region
```

Per ulteriori informazioni sulle modifiche ai parametri e per un elenco dei comandi idonei per la ridenominazione, vedi [Modifiche dei parametri di Redis 5.0.3](#).

- Flussi Redis: questo crea una struttura di dati di registro che consente ai produttori di accodare nuovi elementi in tempo reale. Inoltre consente ai consumatori di utilizzare i messaggi in modo bloccante e non. I flussi consentono i gruppi di consumatori, che rappresentano un gruppo di clienti che consumano cooperativamente diverse parti dello stesso flusso di messaggi, in modo simile ad [Apache Kafka](#). Per ulteriori informazioni, consulta l'argomento relativo all'[introduzione ai flussi Redis](#).
- Support per una famiglia di comandi di flusso come XADD, XRANGE e XREAD. Per ulteriori informazioni, consulta la sezione relativa ai [comandi dei flussi Redis](#).
- Alcuni parametri nuovi e ridenominati. Per ulteriori informazioni, consulta [Modifiche dei parametri di Redis 5.0.0](#).
- Un nuovo parametro Redis, `StreamBasedCmds`.
- Tempo di snapshot leggermente più rapido per i nodi Redis.

⚠ Important

Amazon ElastiCache for Redis ha effettuato il backporting di due correzioni di bug critiche da [Redis open source versione 5.0.1](#). Sono elencati qui di seguito:

- Risposta di mancata corrispondenza RESTORE quando determinate chiavi sono già scadute.
- Il comando XCLAIM può potenzialmente restituire una voce errata o desincronizzare il protocollo.

Entrambe queste correzioni di bug sono incluse nel support ElastiCache for Redis per il motore Redis versione 5.0.0 e verranno utilizzate nei futuri aggiornamenti della versione.

Per ulteriori informazioni, consulta le [note di rilascio di Redis 5.0.6](#) alla pagina Redis di GitHub.

ElastiCache per Redis versione 5.0.5 (obsoleta, usa versione 5.0.6)

Amazon ElastiCache for Redis presenta la nuova versione del motore Redis. Include modifiche alla configurazione online per ElastiCache for Redis di cluster di failover automatico durante tutte le operazioni pianificate. Puoi ora dimensionare il tuo cluster, effettuare l'upgrade della versione del motore Redis e applicare patch e aggiornamenti di manutenzione mentre i cluster rimangono online e continuano a servire le richieste in arrivo. Include anche correzioni di bug.

Per ulteriori informazioni, consulta le [note di rilascio di Redis 5.0.5](#) nella sezione dedicata a Redis su GitHub.

ElastiCache per Redis versione 5.0.4 (obsoleta, usa versione 5.0.6)

Amazon ElastiCache for Redis presenta la nuova versione del motore Redis supportata da Amazon ElastiCache. Include i miglioramenti seguenti:

- Garanzia di stabilità del motore in condizioni speciali.
- Miglioramento della gestione degli errori di Hyperloglog.
- Comandi avanzati di handshake per una replica affidabile.
- Tracciamento costante della consegna dei messaggi tramite il comando XCLAIM.
- Gestione migliorata del campo LFU negli oggetti.
- Gestione migliorata delle transazioni durante l'utilizzo ZPOP.

Per ulteriori informazioni, consulta le [note di rilascio di Redis 5.0.4](#) alla pagina Redis di GitHub.

ElastiCache per Redis versione 5.0.3 (obsoleta, usa versione 5.0.6)

Amazon ElastiCache per Redis introduce la nuova versione del motore Redis supportata da Amazon ElastiCache, che include correzioni di bug.

ElastiCache per Redis versione 5.0.0 (obsoleta, usa versione 5.0.6)

Amazon ElastiCache for Redis presenta la nuova versione principale del motore Redis supportata da Amazon ElastiCache. ElastiCache for Redis 5.0.0 estende il support per i seguenti miglioramenti:

- Flussi Redis: questo crea una struttura di dati di registro che consente ai produttori di accodare nuovi elementi in tempo reale. Inoltre consente ai consumatori di utilizzare i messaggi in modo bloccante e non. I flussi consentono i gruppi di consumatori, che rappresentano un gruppo di clienti che consumano cooperativamente diverse parti dello stesso flusso di messaggi, in modo simile ad [Apache Kafka](#). Per ulteriori informazioni, consulta l'argomento relativo all'[introduzione ai flussi Redis](#).
- Support per una famiglia di comandi di flusso come XADD, XRANGE e XREAD. Per ulteriori informazioni, consulta la sezione relativa ai [comandi dei flussi Redis](#).
- Alcuni parametri nuovi e ridenominati. Per ulteriori informazioni, consulta [Modifiche dei parametri di Redis 5.0.0](#).
- Un nuovo parametro Redis, `StreamBasedCmds`.
- Tempo di snapshot leggermente più rapido per i nodi Redis.

ElastiCache for Redis versione 4.0.10 (potenziata)

Amazon ElastiCache for Redis presenta la nuova versione principale del motore Redis supportata da Amazon ElastiCache. Amazon ElastiCache for Redis 4.0.10 estende il support per i seguenti miglioramenti:

- Ridimensionamento del cluster online e crittografia in una singola versione di ElastiCache for Redis. Per ulteriori informazioni, consulta quanto segue:
 - [Ridimensionamento dei cluster in Redis \(modalità cluster attivata\)](#)
 - [Reshard online e ribilanciamento della partizione per Redis \(modalità cluster abilitata\)](#)
 - [Sicurezza dei dati in Amazon ElastiCache](#)

- Alcuni nuovi parametri. Per ulteriori informazioni, consulta [Modifiche dei parametri di Redis 4.0.10](#).
- Support per la famiglia di comandi di memoria, come MEMORY. Per ulteriori informazioni, consulta [Comandi Redis](#) (cerca MEMO).
- Support per la deframmentazione della memoria in modalità online e, di conseguenza, un utilizzo più efficiente della memoria e più memoria disponibile per i dati.
- Support per svuotamenti ed eliminazioni asincrone. ElastiCache for Redis supporta comandi come UNLINK, FLUSHDB e FLUSHALL per l'esecuzione in un thread diverso dal thread principale. Questo consente di migliorare i tempi di risposta e le prestazioni per le applicazioni liberando la memoria in modo asincrono.
- Un nuovo parametro Redis, ActiveDefragHits. Per ulteriori informazioni, consulta la pagina relativa ai [parametri di Redis](#).

Gli utenti Redis (modalità cluster disabilitata) che eseguono Redis versione 3.2.10 possono avvalersi della console per aggiornare i propri cluster online.

Confronto di ElastiCache per il ridimensionamento del cluster Redis e support crittaggio

Funzionalità	3.2.6	3.2.10	versioni 4.0.10 e successive
Ridimensionamento di cluster online *	No	Sì	Sì
Crittografia dei dati in transito **	Sì	No	Sì
Crittografia dei dati inattivi **	Sì	No	Sì

* Aggiunta, rimozione e ribilanciamento delle partizioni.

** Necessaria per applicazioni conformi a FedRAMP, HIPAA e PCI DSS. Per ulteriori informazioni, consulta [Convalida della conformità per Amazon ElastiCache](#).

Versioni precedenti di fine del ciclo di vita (EOL) (3.x)

ElastiCache for Redis versione 3.2.10 (potenziata)

Amazon ElastiCache for Redis presenta la nuova versione principale del motore Redis supportata da Amazon ElastiCache. ElastiCache for Redis 3.2.10 introduce la funzione di ridimensionamento del

cluster online, per aggiungere o rimuovere partizioni dal cluster mentre continua a servire le richieste di I/O in entrata. Gli utenti di ElastiCache for Redis 3.2.10 dispongono di tutte le caratteristiche delle versioni precedenti di Redis, tranne la possibilità di crittografare i propri dati. Questa caratteristica è attualmente disponibile nella versione 3.2.6.

Confronto tra ElastiCache e Redis versioni 3.2.6 e 3.2.10

Funzionalità	3.2.6	3.2.10
Ridimensionamento di cluster online *	No	Sì
Crittografia dei dati in transito **	Sì	No
Crittografia dei dati inattivi **	Sì	No

* Aggiunta, rimozione e ribilanciamento delle partizioni.

** Necessaria per applicazioni conformi a FedRAMP, HIPAA e PCI DSS. Per ulteriori informazioni, consulta [Convalida della conformità per Amazon ElastiCache](#).

Per ulteriori informazioni, consulta quanto segue:

- [Reshard online e ribilanciamento della partizione per Redis \(modalità cluster abilitata\)](#)
- [Ridimensionamento di cluster online](#)

ElastiCache for Redis versione 3.2.6 (potenziata)

Amazon ElastiCache for Redis presenta la nuova versione principale del motore Redis supportata da Amazon ElastiCache. Gli utenti di ElastiCache for Redis 3.2.6 dispongono di tutte le funzionalità delle versioni precedenti di Redis e possono inoltre crittografare i propri dati. Per ulteriori informazioni, consulta quanto segue:

- [Crittografia in transito ElastiCache \(TLS\)](#)
- [Crittografia dei dati a riposo in ElastiCache](#)
- [Convalida della conformità per Amazon ElastiCache](#)

ElastiCache for Redis versione 3.2.4 (potenziata)

Amazon ElastiCache for Redis versione 3.2.4 propone la nuova versione principale del motore Redis supportata da Amazon ElastiCache. Gli utenti di ElastiCache for Redis 3.2.4 dispongono di tutte le funzionalità delle versioni precedenti di Redis e possono inoltre operare in modalità cluster o non in cluster. La tabella seguente funge da riepilogo.

Confronto tra Redis 3.2.4 in modalità cluster e non in cluster

Funzionalità	Modalità non cluster	Modalità cluster
Partizionamento dei dati	No	Sì
Indicizzazione geospaziale	Sì	Sì
Modifica del tipo di nodo	Sì	Sì *
Dimensionamento delle repliche	Sì	Sì *
Aumento orizzontale	No	Sì *
Support per database	Più di uno	Singolo
Gruppo di parametri	<code>default.redis3.2</code> **	<code>default.redis3.2.cluster.on</code> **

Per informazioni, consulta [Ripristino da un backup in una nuova cache](#)

**O un gruppo da esso derivato.

Note:

- **Partizionamento:** La possibilità di ripartire i dati tra un minimo di 2 e un massimo di 500 gruppi di nodi (partizioni), con il support della replica per ciascun gruppo di nodi.
- **Indicizzazione geospaziale :** Redis 3.2.4 introduce il support per l'indicizzazione geospaziale tramite sei comandi geografici. Per ulteriori informazioni, consulta la documentazione dedicata ai [comandi GEO* di Redis](#), sulla pagina dei comandi Redis (filtrata per GEO).

Per informazioni sulle ulteriori funzioni di Redis 3, consulta le [note di rilascio di Redis 3.2](#) e [note di rilascio di Redis 3.0](#).

Al momento, il Redis gestito da ElastiCache (modalità cluster abilitata) non supporta le seguenti funzioni di Redis 3.2:

- Migrazione di repliche
- Ribilanciamento di cluster
- Debugger Lua

ElastiCache disabilita i seguenti comandi di gestione di Redis 3.2:

- `cluster meet`
- `cluster replicate`
- `cluster flushslots`
- `cluster addslots`
- `cluster delslots`
- `cluster setslot`
- `cluster saveconfig`
- `cluster forget`
- `cluster failover`
- `cluster bumpepoch`
- `cluster set-config-epoch`
- `cluster reset`

Per informazioni sui parametri di Redis 3.2.4, consulta [Modifiche dei parametri di Redis 3.2.4](#).

Versioni precedenti di fine del ciclo di vita (EOL) (2.x)

ElastiCache for Redis versione 2.8.24 (potenziata)

I miglioramenti di Redis a partire dalla versione 2.8.23 includono la correzione di alcuni bug e la registrazione di indirizzi di accesso alla memoria con errori. Per ulteriori informazioni, consulta le [note di rilascio di Redis 2.8](#).

ElastiCache for Redis versione 2.8.23 (potenziata)

Tra i miglioramenti di Redis applicati a partire dalla versione 2.8.22 sono incluse le correzioni di alcuni bug. Per ulteriori informazioni, consulta le [note di rilascio di Redis 2.8](#). Questa versione include, tra l'altro, il support per il nuovo parametro `close-on-slave-write` che, se abilitato, disconnette i client che tentano di scrivere in una replica di sola lettura.

Per ulteriori informazioni sui parametri di Redis 2.8.23, consulta [Parametri aggiunti in Redis 2.8.23 \(avanzato\)](#) nella guida per l'utente di ElastiCache.

ElastiCache for Redis versione 2.8.22 (potenziata)

Tra i miglioramenti di Redis applicati a partire dalla versione 2.8.21 sono inclusi:

- Support per sincronizzazioni e backup senza fork, che permettono di allocare meno memoria per la gestione dei backup e più, invece, per l'applicazione. Per ulteriori informazioni, consulta [Modalità di implementazione di sincronizzazione e backup](#). L'elaborazione senza fork può influire su latenza e throughput. Nel caso di un elevato throughput di scrittura, al momento del riavvio della sincronizzazione e per tutta la sua durata, una replica può non essere raggiungibile.
- In caso di failover, i gruppi di replica ora vengono ripristinati più velocemente perché, se possibile, le repliche eseguono sincronizzazioni parziali anziché complete con il nodo primario. Si guadagna ulteriormente in rapidità perché, inoltre, sia il primario che le repliche non utilizzano più il disco durante le sincronizzazioni.
- Support per due nuovi parametri CloudWatch.
 - `ReplicationBytes`:- Il numero di byte che il cluster primario di un gruppo di replica invia alle repliche di lettura.
 - `SaveInProgress` - Un valore binario che indica se sia in esecuzione o meno un processo di salvataggio in background.

Per ulteriori informazioni, consulta [Monitoraggio dell'utilizzo con i parametri di CloudWatch](#).

- Varie correzioni di bug critici nel comportamento della PSYNC di replica. Per ulteriori informazioni, consulta le [note di rilascio di Redis 2.8](#).
- Per mantenere elevato il livello di prestazioni delle repliche nei gruppi di replica con la funzione Multi-AZ e per una maggiore stabilità del cluster, le repliche non di ElastiCache non sono più supportate.
- Per affinare la coerenza dei dati tra il cluster primario e le repliche in un gruppo di replica, le repliche non eliminano più le chiavi indipendenti dal cluster principale.

- Le variabili di configurazione Redis `appendonly` e `appendfsync` non sono supportate su Redis versione 2.8.22 e successive.
- In caso di memoria insufficiente, i client con un buffer di output consistente possono essere disconnessi da un cluster di replica. Se disconnesso, il client deve riconnettersi. Situazioni simili si verificano più frequentemente con i client PUBSUB.

ElastiCache for Redis versione 2.8.21

Tra i miglioramenti di Redis applicati a partire dalla versione 2.8.19 sono incluse varie correzioni di bug. Per ulteriori informazioni, consulta le [note di rilascio di Redis 2.8](#).

ElastiCache for Redis versione 2.8.19

Tra i miglioramenti di Redis applicati a partire dalla versione 2.8.6 sono inclusi:

- Support per HyperLogLog. Per ulteriori informazioni in merito, consulta il documento dedicato alla [nuova struttura di dati Redis: HyperLogLog](#).
- I dati basati su set ordinati estendono il support alle query di intervallo lessicografiche con i nuovi comandi ZRANGEBYLEX, ZLEXCOUNT e ZREMRANGEBYLEX.
- Per impedire che un nodo primario invii dati non aggiornati ai nodi di replica, la sincronizzazione del master si blocca quando un processo di salvataggio in background (`bgsave`) figlio si interrompe.
- Support per il parametro `HyperLogLogBasedCommands` della metrica CloudWatch . Per ulteriori informazioni, consulta [Parametri di Redis](#).

ElastiCache for Redis versione 2.8.6

Tra i miglioramenti di Redis applicati a partire dalla versione 2.6.13 sono inclusi:

- Resilienza e tolleranza ai guasti migliorate per le repliche di lettura.
- Support per la risincronizzazione parziale.
- Support per un numero minimo definito dall'utente di repliche di lettura che devono essere sempre disponibili.
- Support completo per pub/sub-notifica ai client degli eventi sul server.
- Rilevamento automatico di un errore del nodo primario e failover di quest'ultimo con un nodo secondario.

ElastiCache for Redis versione 2.6.13

Redis 2.6.13 è la prima versione di Redis supportata da Amazon ElastiCache for Redis. La funzione Multi-AZ non è supportata su Redis 2.6.13.

Pianificazione della fine del ciclo di vita delle versioni di Redis


Questa sezione definisce le date di fine vita (EOL) per le versioni principali precedenti mano a mano che vengono annunciate. Ciò consente di prendere decisioni future sulla versione e sugli aggiornamenti.


Note

Le versioni delle patch di ElastiCache per Redis dalla 5.0.0 alla 5.0.5 sono obsolete. Usa versione 5.0.6 o successiva.

La tabella seguente riepiloga ogni versione e la relativa data di fine vita annunciata, nonché la versione di destinazione dell'aggiornamento consigliata.

EOL precedenti

V p e d o	Versioni secondarie di origine	Destinazione aggiornamento consigliata	Data di fine vita
V 3	3.2.4, 3.2.6 e 3.2.10	Versione 6.2 o successiva  Note Per le regioni US-ISO-EAST-1, US-ISO-WEST-1 e US-ISOB-EAST-1, è preferibile la versione 5.0.6 o superiore.	31 luglio 2023

Versioni secondarie di origine	Destinazione aggiornamento consigliata	Data di fine vita
V p e d o V 2,8,24, 2,8,23, 2,8,22, 2,8,21, 2 2,8,19, 2,8,12, 2,8,6, 2,6,13	Versione 6.2 o successiva <div data-bbox="613 495 1042 898"><p> Note</p><p>Per le regioni US-ISO-EAST-1, US-ISO-WEST-1 e US-ISOB-EAST-1, è preferibile la versione 5.0.6 o superiore.</p></div>	13 gennaio 2023

Come aggiornare la versione di un motore

L'utente può eseguire l'aggiornamento a una versione sul cluster o gruppo di replica, modificando quest'ultimo con la console ElastiCache, lo AWS CLI o l'API di ElastiCache e specificando la nuova versione del motore da utilizzare. Per ulteriori informazioni, consulta i seguenti argomenti.

Come modificare cluster e gruppi di replica	
Cluster	Gruppi di replica
Utilizzando il AWS Management Console	Utilizzo di AWS Management Console
Utilizzando il AWS CLI	Utilizzo di AWS CLI
Utilizzo dell' ElastiCache API	Utilizzo dell'API ElastiCache

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

Per risolvere il blocco di un aggiornamento del motore Redis

- Esegui una di queste operazioni:
 - Programmare l'operazione di aggiornamento del motore Redis in una finestra di manutenzione successiva, deselegionando la casella di controllo Apply immediately (Applica immediatamente).

Con la CLI utilizzare `--no-apply-immediately`. Con l'API, utilizzare `ApplyImmediately=false`.

- Attendere una finestra di manutenzione successiva per l'esecuzione dell'aggiornamento del motore Redis.
- Aggiungere l'operazione di dimensionamento Redis alle modifiche del cluster con la casella di controllo Apply Immediately (Applica immediatamente) selezionata.

Con la CLI utilizzare `--apply-immediately`. Con l'API, utilizzare `ApplyImmediately=true`.

Questo approccio annulla efficacemente l'aggiornamento del motore nella finestra di manutenzione successiva, poiché lo si esegue immediatamente.

Differenze di compatibilità e comportamento delle versioni principali

Important

La pagina seguente è strutturata per indicare tutte le differenze di incompatibilità tra le versioni e informare l'utente di eventuali considerazioni da fare durante l'aggiornamento alle versioni più recenti. Questo elenco include tutti i problemi di incompatibilità delle versioni che potrebbero verificarsi durante l'aggiornamento.

Puoi eseguire l'aggiornamento direttamente dalla versione Redis corrente all'ultima versione Redis disponibile, senza la necessità di aggiornamenti sequenziali. Ad esempio, puoi eseguire l'aggiornamento direttamente da Redis versione 3.0 a Redis versione 7.0.

Le versioni di Redis sono identificate con una versione semantica che comprende un componente MAJOR, MINOR e PATCH. Ad esempio, in Redis 4.0.10 la versione principale è 4, la versione secondaria 0 e la versione patch 10. Questi valori generalmente vengono incrementati in base alle convenzioni seguenti:

- Le versioni MAJOR sono per modifiche incompatibili con l'API
- Le versioni MINOR sono per nuove funzionalità aggiunte per compatibilità a ritroso
- Le versioni PATCH sono per correzioni di bug compatibili a ritroso e modifiche non funzionali

È preferibile rimanere sempre sull'ultima versione della patch in una determinata versione MAJOR.MINOR, per sfruttare gli ultimi miglioramenti in termini di prestazioni e stabilità. A partire da Redis 6.0, ElastiCache per Redis offrirà una versione singola per ogni release principale Redis OSS, piuttosto che più versioni secondarie. ElastiCache per Redis gestirà automaticamente la versione secondaria dei cluster di cache in esecuzione, garantendo prestazioni migliorate e maggiore sicurezza.

È preferibile, inoltre, eseguire periodicamente l'aggiornamento all'ultima versione principale, siccome la maggior parte dei miglioramenti principali non viene ripristinata alle versioni precedenti. Quando ElastiCache estende la disponibilità a una nuova regione AWS, ElastiCache per Redis supporta per la nuova regione le due versioni MAJOR.MINOR più recenti in quel momento. Ad esempio, se viene lanciata una nuova regione AWS e le ultime versioni di MAJOR.MINOR ElastiCache per Redis sono 7.0 e 6.2, ElastiCache per Redis supporterà le versioni 7.0 e 6.2 nella nuova regione AWS. Man mano che verranno rilasciate le nuove versioni MAJOR.MINOR di ElastiCache per Redis, ElastiCache continuerà ad aggiungere il supporto per le nuove versioni di ElastiCache per Redis. Per ulteriori informazioni sulla scelta delle regioni per ElastiCache, consulta [Scelta di regioni e zone di disponibilità](#).

Quando esegui un aggiornamento che include versioni principali o secondarie, considerare l'elenco seguente che include il comportamento e le modifiche incompatibili a ritroso rilasciate con Redis nel tempo.

Modifiche del comportamento di Redis 7.0 e modifiche non retrocompatibili

Per l'elenco completo delle modifiche, consulta [Note di rilascio di Redis 7.0](#).

- `SCRIPT LOAD` e `SCRIPT FLUSH` non sono più propagati alle repliche. Se occorre durabilità degli script, consigliamo di valutare l'uso delle [funzioni Redis](#).
- I canali Pubsub sono ora bloccati per impostazione predefinita per i nuovi utenti ACL.
- Il comando `STRALGO` è stato sostituito con il comando `LCS`.
- Il formato per `ACL GETUSER` è stato modificato in modo che tutti i campi contengano il modello di stringa di accesso standard. Se l'automazione era dovuta all'utilizzo di `ACL GETUSER`, occorre verificare che siano gestiti entrambi i formati.
- Le categorie ACL per `SELECT`, `WAIT`, `ROLE`, `LASTSAVE`, `READONLY`, `READWRITE` e `ASKING` sono cambiate.
- Il comando `INFO` mostra ora le statistiche per sottocomando anziché nei comandi del container del livello superiore.

- I valori restituiti dai comandi LPOP, RPOP, ZPOPMIN e ZPOPMAX sono cambiati in determinati casi limite. Se si utilizzano questi comandi, occorre controllare le note di rilascio e valutare se hanno un impatto.
- I comandi SORT e SORT_RO richiedono ora l'accesso all'intero keyspace per poter utilizzare gli argomenti GET e BY.

Modifiche del comportamento di Redis 6.2 e modifiche non compatibili a ritroso

Per l'elenco completo delle modifiche, consulta [Note di rilascio di Redis 6.2](#).

- I flag ACL dei comandi TIME, ECHO, ROLE e LASTSAVE sono stati modificati. Ciò può causare il rifiuto di comandi precedentemente autorizzati e viceversa.

Note

Nessuno di questi comandi modifica o fornisce accesso ai dati.

- Quando esegui l'aggiornamento da Redis 6.0, l'ordine delle coppie chiave/valore restituite da una risposta mappa a uno script lua viene modificato. Se i tuoi script utilizzano `redis.setresp()` o restituiscono una mappa (novità in Redis 6.0), considera il rischio che lo script potrebbe interrompersi con gli aggiornamenti.

Modifiche del comportamento di Redis 6.0 e modifiche non compatibili a ritroso

Per l'elenco completo delle modifiche, consulta [Note di rilascio di Redis 6.0](#).

- Il numero massimo di database consentiti è stato ridotto da 1,2 milioni a 10.000. Il valore predefinito è 16 e sconsigliamo di utilizzare valori molto più grandi di questo poiché abbiamo riscontrato problemi di prestazioni e memoria.
- Imposta il parametro `AutoMinorVersionUpgrade` su `yes`: ElastiCache per Redis gestirà l'aggiornamento delle versioni secondarie tramite aggiornamenti self-service. Questa operazione verrà gestita tramite canali standard di notifica dei clienti tramite una campagna di aggiornamento self-service. Per ulteriori informazioni, consulta [Aggiornamenti self-service in ElastiCache](#).

Modifiche del comportamento di Redis 5.0 e modifiche non compatibili a ritroso

Per l'elenco completo delle modifiche, consulta [Note di rilascio di Redis 5.0](#).

- Gli script vengono replicati dagli effetti invece di rieseguire lo script sulla replica. Ciò generalmente migliora le prestazioni, ma può aumentare la quantità di dati replicati tra primari e repliche. Esiste un'opzione per tornare al comportamento precedente, disponibile solo in ElastiCache per Redis 5.0.
- Se stai eseguendo l'aggiornamento da Redis 4.0, alcuni comandi negli script LUA restituiranno gli argomenti in un ordine diverso rispetto a quello delle versioni precedenti. In Redis 4.0, Redis ordina alcune risposte lessograficamente per rendere le risposte deterministiche; tale ordinamento non viene applicato quando gli script vengono replicati dagli effetti.
- A partire da Redis 5.0.3 e versioni superiori, ElastiCache per Redis scaricherà alcune operazioni I/O su core in background su tipi di istanza con oltre 4 vCPU. Ciò potrebbe modificare le caratteristiche delle prestazioni Redis e cambiare i valori di alcune metriche. Per ulteriori informazioni, consulta [Quali parametri è opportuno monitorare?](#) per appurare se è necessario modificare le metriche.

Modifiche del comportamento di Redis 4.0 e modifiche non compatibili a ritroso

Per l'elenco completo delle modifiche, consulta [Note di rilascio di Redis 4.0](#).

- Il registro lento ora registra altri due argomenti, il nome e l'indirizzo del client. Questa modifica dovrebbe essere compatibile con le versioni precedenti a meno che non si faccia esplicitamente affidamento su ogni voce del registro lento contenente 3 valori.
- Il comando CLUSTER NODES ora restituisce un formato lievemente diverso, non compatibile a ritroso. È preferibile che i client non utilizzino questo comando per conoscere i nodi presenti in un cluster, utilizzando invece CLUSTER SLOTS.

EOL precedenti

Modifiche del comportamento di Redis 3.2 e modifiche non compatibili a ritroso

Per l'elenco completo delle modifiche, consulta [Note di rilascio di Redis 3.2](#).

- Non esistono modifiche di compatibilità da richiamare per questa versione.

Per ulteriori informazioni, consulta [Pianificazione della fine del ciclo di vita delle versioni di Redis](#).

Modifiche del comportamento di Redis 2.8 e modifiche non compatibili a ritroso

Per l'elenco completo delle modifiche, consulta [Note di rilascio di Redis 2.8](#).

- A partire da Redis 2.8.22, Redis AOF non è più supportato in ElastiCache per Redis. È preferibile utilizzare MemoryDB quando i dati devono essere conservati in modo duraturo.
- A partire da Redis 2.8.22, ElastiCache per Redis non supporta più il collegamento di repliche ai principali ospitati in ElastiCache. Durante l'aggiornamento, le repliche esterne verranno scollegate e non potranno ricollegarsi. È preferibile utilizzare la cache lato client, resa disponibile in Redis 6.0 come alternativa alle repliche esterne.
- I comandi TTL e PTTL ora restituiscono -2 se la chiave non esiste e -1 se esiste ma non ha una scadenza associata. Redis 2.6 e versioni precedenti restituivano -1 per entrambe le condizioni.
- SORT con ALPHA ora ordina in base alle impostazioni locali di confronto se non viene utilizzata alcuna opzione STORE.

Per ulteriori informazioni, consulta [Pianificazione della fine del ciclo di vita delle versioni di Redis](#).

Best practice e strategie di caching ElastiCache

Di seguito sono indicate le best practice consigliate per l'utilizzo di Amazon ElastiCache. Queste best practice consentono di migliorare prestazioni e affidabilità della cache.

Argomenti

- [Utilizzo di Redis](#)
- [Best practice con i client Redis](#)
- [Best practice per l'utilizzo di cluster progettati autonomamente](#)
- [Best practice per Redis](#)
- [Strategie di caching](#)

Utilizzo di Redis

Di seguito sono riportate informazioni sull'interfaccia Redis in ElastiCache.

Argomenti

- [Comandi Redis supportati e limitati](#)
- [Configurazione e limiti di Redis](#)

Comandi Redis supportati e limitati

Comandi Redis supportati

Comandi Redis supportati

I seguenti comandi Redis sono supportati dalle cache serverless. Oltre a questi comandi, sono supportati anche i [Comandi JSON di Redis non supportati](#).

Comandi bitmap

- BITCOUNT

Conta il numero di bit impostati (conteggio della popolazione) in una stringa.

[Ulteriori informazioni](#)

- BITFIELD

Esegue operazioni arbitrarie di numeri interi bitfield sulle stringhe.

[Ulteriori informazioni](#)

- BITFIELD_RO

Esegue operazioni arbitrarie di numeri interi bitfield di sola lettura sulle stringhe.

[Ulteriori informazioni](#)

- BITOP

Esegue operazioni bit per bit su più stringhe e archivia il risultato.

[Ulteriori informazioni](#)

- BITPOS

Trova il primo bit di impostazione (1) o cancellazione (0) in una stringa.

[Ulteriori informazioni](#)

- GETBIT

Restituisce un valore di bit per offset.

[Ulteriori informazioni](#)

- SETBIT

Imposta o cancella il bit all'offset del valore della stringa. Crea la chiave, se non esiste.

[Ulteriori informazioni](#)

Comandi di gestione dei cluster

- CLUSTER COUNTKEYSINSLOT

Restituisce il numero di chiavi in uno slot hash.

[Ulteriori informazioni](#)

- CLUSTER GETKEYSINSLOT

Restituisce i nomi delle chiavi in uno slot hash.

[Ulteriori informazioni](#)

- CLUSTER INFO

Restituisce le informazioni sullo stato di un nodo. In una cache serverless, restituisce lo stato della singola "partizione" virtuale esposta al client.

[Ulteriori informazioni](#)

- CLUSTER KEYSLOT

Restituisce lo slot hash per una chiave.

[Ulteriori informazioni](#)

- CLUSTER MYID

Restituisce l'ID di un nodo. In una cache serverless, restituisce lo stato della singola "partizione" virtuale esposta al client.

[Ulteriori informazioni](#)

- CLUSTER NODES

Restituisce la configurazione del cluster per un nodo. In una cache serverless, restituisce lo stato della singola "partizione" virtuale esposta al client.

[Ulteriori informazioni](#)

- CLUSTER REPLICAS

Elenca i nodi di replica di un nodo principale. In una cache serverless, restituisce lo stato della singola "partizione" virtuale esposta al client.

[Ulteriori informazioni](#)

- CLUSTER SHARDS

Restituisce la mappatura degli slot del cluster alle partizioni. In una cache serverless, restituisce lo stato della singola "partizione" virtuale esposta al client.

[Ulteriori informazioni](#)

- CLUSTER SLOTS

Restituisce la mappatura degli slot del cluster ai nodi. In una cache serverless, restituisce lo stato della singola "partizione" virtuale esposta al client.

[Ulteriori informazioni](#)

- READONLY

Abilita le query di sola lettura per una connessione a un nodo di replica del cluster Redis.

[Ulteriori informazioni](#)

- READWRITE

Abilita le query di lettura-scrittura per una connessione a un nodo di replica del cluster Redis.

[Ulteriori informazioni](#)

Comandi di gestione della connessione

- AUTH

Autentica la connessione.

[Ulteriori informazioni](#)

- CLIENT GETNAME

Restituisce il nome della connessione.

[Ulteriori informazioni](#)

- CLIENT REPLY

Indica al server se rispondere ai comandi.

[Ulteriori informazioni](#)

- CLIENT SETNAME

Imposta il nome della connessione.

[Ulteriori informazioni](#)

- ECHO

Restituisce la specifica stringa.

[Ulteriori informazioni](#)

- HELLO

Esegue l'handshake con il server Redis.

[Ulteriori informazioni](#)

- PING

Restituisce la risposta di attività del server.

[Ulteriori informazioni](#)

- QUIT

Chiude la connessione.

[Ulteriori informazioni](#)

- RESET

Reimposta la connessione.

[Ulteriori informazioni](#)

- SELECT

Cambia il database selezionato.

[Ulteriori informazioni](#)

Comandi generici

- COPY

Copia il valore di una chiave in una nuova chiave.

[Ulteriori informazioni](#)

- DEL

Elimina una o più chiavi.

[Ulteriori informazioni](#)

- DUMP

Restituisce una rappresentazione serializzata del valore archiviato in una chiave.

[Ulteriori informazioni](#)

- EXISTS

Determina se esistono una o più chiavi.

[Ulteriori informazioni](#)

- EXPIRE

Imposta l'ora di scadenza di una chiave in secondi.

[Ulteriori informazioni](#)

- EXPIREAT

Imposta l'ora di scadenza di una chiave su un timestamp Unix.

[Ulteriori informazioni](#)

- EXPIRETIME

Restituisce l'ora di scadenza di una chiave come timestamp Unix.

[Ulteriori informazioni](#)

- PERSIST

Rimuove l'ora di scadenza di una chiave.

[Ulteriori informazioni](#)

- PEXPIRE

Imposta l'ora di scadenza di una chiave in millisecondi.

[Ulteriori informazioni](#)

- PEXPIREAT

Imposta l'ora di scadenza di una chiave su un timestamp in millisecondi Unix.

[Ulteriori informazioni](#)

- PEXPIRETIME

Restituisce l'ora di scadenza di una chiave come timestamp in millisecondi Unix.

[Ulteriori informazioni](#)

- PTTL

Restituisce l'ora di scadenza di una chiave in millisecondi.

[Ulteriori informazioni](#)

- RANDOMKEY

Restituisce un nome di chiave casuale del database.

[Ulteriori informazioni](#)

- RENAME

Rinomina una chiave e sovrascrive la destinazione.

[Ulteriori informazioni](#)

- RENAMENX

Rinomina una chiave solo quando il nome della chiave di destinazione non esiste.

[Ulteriori informazioni](#)

- RESTORE

Crea una chiave dalla rappresentazione serializzata di un valore.

[Ulteriori informazioni](#)

- SCAN

Esegue l'iterazione dei nomi delle chiavi nel database.

[Ulteriori informazioni](#)

- SORT

Ordina gli elementi in un elenco, un set o un set ordinato e archivia facoltativamente il risultato.

[Ulteriori informazioni](#)

- SORT_RO

Restituisce gli elementi ordinati di un elenco, un set o un set ordinato.

[Ulteriori informazioni](#)

- TOUCH

Restituisce il numero di chiavi esistenti tra quelle specificate dopo l'aggiornamento dell'ora dell'ultimo accesso.

[Ulteriori informazioni](#)

- TTL

Restituisce l'ora di scadenza di una chiave in secondi.

[Ulteriori informazioni](#)

- TYPE

Determina il tipo di valore archiviato in una chiave.

[Ulteriori informazioni](#)

- UNLINK

Elimina in modo asincrono una o più chiavi.

[Ulteriori informazioni](#)

Comandi geospaziali

- GEOADD

Aggiunge uno o più membri a un indice geospaziale. La chiave viene creata se non esiste.

[Ulteriori informazioni](#)

- GEODIST

Restituisce la distanza tra due membri di un indice geospaziale.

[Ulteriori informazioni](#)

- GEOHASH

Restituisce i membri di un indice geospaziale come stringhe geohash.

[Ulteriori informazioni](#)

- GEOPOS

Restituisce la longitudine e la latitudine dei membri di un indice geospaziale.

[Ulteriori informazioni](#)

- GEORADIUS

Esegue le query su un indice geospaziale per i membri che si trovano entro una distanza da una coordinata e facoltativamente archivia il risultato.

[Ulteriori informazioni](#)

- GEORADIUS_RO

Restituisce i membri di un indice geospaziale che si trovano entro una distanza da una coordinata.

[Ulteriori informazioni](#)

- GEORADIUSBYMEMBER

Esegue le query su un indice geospaziale per i membri che si trovano entro una distanza da un membro e facoltativamente archivia il risultato.

[Ulteriori informazioni](#)

- GEORADIUSBYMEMBER_RO

Restituisce i membri di un indice geospaziale che si trovano entro una distanza da un membro.

[Ulteriori informazioni](#)

- GEOSEARCH

Esegue le query su un indice geospaziale per i membri presenti all'interno dell'area di un riquadro o di un cerchio.

[Ulteriori informazioni](#)

- GEOSEARCHSTORE

Esegue le query su un indice geospaziale per i membri presenti all'interno dell'area di un riquadro o di un cerchio e facoltativamente archivia il risultato.

[Ulteriori informazioni](#)

Comandi hash

- HDEL

Elimina da un hash uno o più campi e i relativi valori. Elimina l'hash se non rimane alcun campo.

[Ulteriori informazioni](#)

- HEXISTS

Determina se un campo è presente in un hash.

[Ulteriori informazioni](#)

- HGET

Restituisce il valore di un campo in un hash.

[Ulteriori informazioni](#)

- HGETALL

Restituisce tutti i campi e i valori in un hash.

[Ulteriori informazioni](#)

- HINCRBY

Incrementa di un numero il valore intero di un campo in un hash. Utilizza 0 come valore iniziale se il campo non esiste.

[Ulteriori informazioni](#)

- HINCRBYFLOAT

Incrementa di un numero il valore in virgola mobile di un campo. Utilizza 0 come valore iniziale se il campo non esiste.

[Ulteriori informazioni](#)

- HKEYS

Restituisce tutti i campi in un hash.

[Ulteriori informazioni](#)

- HLEN

Restituisce il numero di campi in un hash.

[Ulteriori informazioni](#)

- HMGET

Restituisce i valori di tutti i campi in un hash.

[Ulteriori informazioni](#)

- HMSET

Imposta i valori di più campi.

[Ulteriori informazioni](#)

- HRANDFIELD

Restituisce uno o più campi casuali di un hash.

[Ulteriori informazioni](#)

- HSCAN

Esegue l'iterazione dei campi e dei valori di un hash.

[Ulteriori informazioni](#)

- HSET

Crea o modifica il valore di un campo in un hash.

[Ulteriori informazioni](#)

- HSETNX

Imposta il valore di un campo in un hash solo quando il campo non esiste.

[Ulteriori informazioni](#)

- HSTRLEN

Restituisce la lunghezza del valore di un campo.

[Ulteriori informazioni](#)

- HVALS

Restituisce tutti i valori in un hash.

[Ulteriori informazioni](#)

Comandi HyperLogLog

- PFADD

Aggiunge elementi a una chiave HyperLogLog. Crea la chiave, se non esiste.

[Ulteriori informazioni](#)

- PFCOUNT

Restituisce la cardinalità approssimativa dei set osservati dalle chiavi HyperLogLog.

[Ulteriori informazioni](#)

- **PFMERGE**

Unisce uno o più valori HyperLogLog in un'unica chiave.

[Ulteriori informazioni](#)

Comandi di elenco

- **BLMOVE**

Visualizza l'elemento di un elenco, lo inserisce in un altro elenco e lo restituisce. Si blocca finché un elemento non è disponibile. Elimina l'elenco se l'ultimo elemento è stato spostato.

[Ulteriori informazioni](#)

- **BLMPOP**

Visualizza il primo elemento da uno degli elenchi. Si blocca finché un elemento non è disponibile. Elimina l'elenco se l'ultimo elemento è stato visualizzato.

[Ulteriori informazioni](#)

- **BLPOP**

Rimuove e restituisce il primo elemento di un elenco. Si blocca finché un elemento non è disponibile. Elimina l'elenco se l'ultimo elemento è stato visualizzato.

[Ulteriori informazioni](#)

- **BRPOP**

Rimuove e restituisce l'ultimo elemento di un elenco. Si blocca finché un elemento non è disponibile. Elimina l'elenco se l'ultimo elemento è stato visualizzato.

[Ulteriori informazioni](#)

- **BRPOPLPUSH**

Visualizza l'elemento di un elenco, lo inserisce in un altro elenco e lo restituisce. Si blocca finché un elemento non è disponibile. Elimina l'elenco se l'ultimo elemento è stato visualizzato.

[Ulteriori informazioni](#)

- **LINDEX**

Restituisce un elemento di un elenco in base all'indice.

[Ulteriori informazioni](#)

- LINSERT

Inserisce un elemento prima o dopo un altro elemento di un elenco.

[Ulteriori informazioni](#)

- LLEN

Restituisce la lunghezza di un elenco.

[Ulteriori informazioni](#)

- LMOVE

Restituisce l'elemento di un elenco dopo averlo visualizzato e spostato in un altro. Elimina l'elenco se l'ultimo elemento è stato spostato.

[Ulteriori informazioni](#)

- LMPOP

Restituisce più elementi di un elenco dopo averli rimossi. Elimina l'elenco se l'ultimo elemento è stato visualizzato.

[Ulteriori informazioni](#)

- LPOP

Restituisce i primi elementi di un elenco dopo averlo rimosso. Elimina l'elenco se l'ultimo elemento è stato visualizzato.

[Ulteriori informazioni](#)

- LPOS

Restituisce l'indice degli elementi corrispondenti di un elenco.

[Ulteriori informazioni](#)

- LPUSH

Antepone uno o più elementi a un elenco. Crea la chiave, se non esiste.

[Ulteriori informazioni](#)

- LPUSHX

Antepone uno o più elementi a un elenco solo quando l'elenco esiste.

[Ulteriori informazioni](#)

- LRANGE

Restituisce un intervallo di elementi di un elenco.

[Ulteriori informazioni](#)

- LREM

Rimuove gli elementi da un elenco. Elimina l'elenco se l'ultimo elemento è stato rimosso.

[Ulteriori informazioni](#)

- LSET

Imposta il valore di un elemento in un elenco in base al relativo indice.

[Ulteriori informazioni](#)

- LTRIM

Rimuove gli elementi all'inizio e alla fine di un elenco. Elimina l'elenco se tutti gli elementi sono stati tagliati.

[Ulteriori informazioni](#)

- RPOP

Restituisce e rimuove gli ultimi elementi di un elenco. Elimina l'elenco se l'ultimo elemento è stato visualizzato.

[Ulteriori informazioni](#)

- RPOPLPUSH

Restituisce l'ultimo elemento di un elenco dopo averlo rimosso e inserito in un altro elenco. Elimina l'elenco se l'ultimo elemento è stato visualizzato.

[Ulteriori informazioni](#)

- R PUSH

Aggiunge uno o più elementi a un elenco. Crea la chiave, se non esiste.


[Ulteriori informazioni](#)

- R PUSHX

Aggiunge un elemento a un elenco solo quando l'elenco esiste.

[Ulteriori informazioni](#)

Comandi Pub/Sub

 Note

I comandi PUBSUB utilizzano internamente PUBSUB partizionato, pertanto i nomi dei canali sono combinati.

- PUBLISH

Invia un messaggio a un canale.

[Ulteriori informazioni](#)

- PUBSUB CHANNELS

Restituisce i canali attivi.

[Ulteriori informazioni](#)

- PUBSUB NUMSUB

Restituisce il numero di abbonati ai canali.

[Ulteriori informazioni](#)

- PUBSUB SHARDCHANNELS

Restituisce i canali di partizione attivi.

[PUBSUB-SHARDCHANNELS](#)

- PUBSUB SHARDNUMSUB

Restituisce il numero di abbonati ai canali di partizione.

[PUBSUB-SHARDNUMSUB](#)

- SPUBLISH

Invia un messaggio a un canale di partizione.

[Ulteriori informazioni](#)

- SSUBSCRIBE

Ascolta i messaggi pubblicati sui canali di partizione.

[Ulteriori informazioni](#)

- SUBSCRIBE

Ascolta i messaggi pubblicati sui canali.

[Ulteriori informazioni](#)

- SUNSUBSCRIBE

Interrompe l'ascolto dei messaggi pubblicati sui canali di partizione.

[Ulteriori informazioni](#)

- UNSUBSCRIBE

Interrompe l'ascolto dei messaggi pubblicati sui canali.

[Ulteriori informazioni](#)

Comandi di scripting

- EVAL

Esegue uno script Lua lato server.

[Ulteriori informazioni](#)

- EVAL_RO

Esegue uno script Lua lato server di sola lettura.

[Ulteriori informazioni](#)

- EVALSHA

Esegue uno script Lua lato server tramite il digest SHA1.

[Ulteriori informazioni](#)

- EVALSHA_RO

Esegue uno script Lua lato server di sola lettura tramite il digest SHA1.

[Ulteriori informazioni](#)

- SCRIPT EXISTS

Determina se esistono script Lua lato server nella cache degli script.

[Ulteriori informazioni](#)

- SCRIPT FLUSH

Attualmente la cache degli script no-op è gestita dal servizio.

[Ulteriori informazioni](#)

- SCRIPT LOAD

Carica uno script Lua lato server nella cache degli script.

[Ulteriori informazioni](#)

Comandi di gestione del server

- ACL CAT

Elenca le categorie ACL o i comandi all'interno di una categoria.

[Ulteriori informazioni](#)

- ACL GENPASS

Genera una password pseudocasuale e sicura che può essere utilizzata per identificare gli utenti ACL.

[Ulteriori informazioni](#)

- ACL GETUSER

Elenca le regole ACL di un utente.

[Ulteriori informazioni](#)

- ACL LIST

Esegue il dump delle regole valide nel formato di file ACL.

[Ulteriori informazioni](#)

- ACL USERS

Elenca tutti gli utenti ACL.

[Ulteriori informazioni](#)

- ACL WHOAMI

Restituisce il nome utente autenticato della connessione corrente.

[Ulteriori informazioni](#)

- DBSIZE

Restituisce il numero di chiavi nel database attualmente selezionato. Non si garantisce che questa operazione sia atomica su tutti gli slot.

[Ulteriori informazioni](#)

- COMMAND

Restituisce informazioni dettagliate su tutti i comandi.

[Ulteriori informazioni](#)

- COMMAND COUNT

Restituisce un numero di comandi.

[Ulteriori informazioni](#)

- COMMAND DOCS

Restituisce informazioni documentarie su uno, molteplici o tutti i comandi.

[Ulteriori informazioni](#)

- COMMAND GETKEYS

Estrae i nomi delle chiavi da un comando arbitrario.

[Ulteriori informazioni](#)

- COMMAND GETKEYSANDFLAGS

Estrae i nomi delle chiavi e i flag di accesso per un comando arbitrario.

[Ulteriori informazioni](#)

- COMMAND INFO

Restituisce informazioni su uno, molteplici o tutti i comandi.

[Ulteriori informazioni](#)

- COMMAND LIST

Restituisce un elenco di nomi di comando.

[Ulteriori informazioni](#)

- FLUSHALL

Rimuove tutte le chiavi da tutti i database. Non si garantisce che questa operazione sia atomica su tutti gli slot.

[Ulteriori informazioni](#)

- FLUSHDB

Rimuove tutte le chiavi dal database corrente. Non si garantisce che questa operazione sia atomica su tutti gli slot.

[Ulteriori informazioni](#)

- INFO

Restituisce informazioni e statistiche sul server.

[Ulteriori informazioni](#)

- LOLWUT

Visualizza la computer art e la versione Redis.

[Ulteriori informazioni](#)

- ROLE

Restituisce il ruolo di replica.

[Ulteriori informazioni](#)

- TIME

Restituisce l'ora del server.

[Ulteriori informazioni](#)

Comandi di set

- SADD

Aggiunge uno o più membri a un set. Crea la chiave, se non esiste.

[Ulteriori informazioni](#)

- SCARDT

Restituisce il numero di membri di un set.

[Ulteriori informazioni](#)

- SDIFF

Restituisce la differenza di più set.

[Ulteriori informazioni](#)

- SDIFFSTORE

Archivia la differenza di più set in una chiave.

[Ulteriori informazioni](#)

- SINTER

Restituisce l'intersezione di più set.

[Ulteriori informazioni](#)

- SINTERCARD

Restituisce il numero di membri dell'intersezione di più set.

[Ulteriori informazioni](#)

- SINTERSTORE

Archivia l'intersezione di più set in una chiave.

[Ulteriori informazioni](#)

- SISMEMBER

Determina se un membro appartiene a un set.

[Ulteriori informazioni](#)

- SMEMBERS

Restituisce tutti i membri di un set.

[Ulteriori informazioni](#)

- SMISMEMBER

Determina se più membri appartengono a un set.

[Ulteriori informazioni](#)

- SMOVE

Sposta un membro da un set all'altro.

[Ulteriori informazioni](#)

- SPOP

Restituisce uno o più membri casuali di un set dopo averli rimossi. Elimina il set se l'ultimo membro è stato visualizzato.

[Ulteriori informazioni](#)

- SRANDMEMBER

Ottiene uno o più membri casuali da un set.

[Ulteriori informazioni](#)

- SREM

Rimuove uno o più membri da un set. Elimina il set se l'ultimo membro è stato rimosso.

[Ulteriori informazioni](#)

- SSCAN

Esegue l'iterazione dei membri di un set.

[Ulteriori informazioni](#)

- SUNION

Restituisce l'unione di più set.

[Ulteriori informazioni](#)

- SUNIONSTORE

Archivia l'unione di più set in una chiave.

[Ulteriori informazioni](#)

Comandi di set ordinati

- BZMPOP

Rimuove e restituisce un membro in base al punteggio da uno o più set ordinati. Si blocca finché un membro non è disponibile. Elimina il set ordinato se l'ultimo elemento è stato visualizzato.

[Ulteriori informazioni](#)

- BZPOPMAX

Rimuove e restituisce il membro con il punteggio più alto da uno o più set ordinati. Si blocca finché un membro non è disponibile. Elimina il set ordinato se l'ultimo elemento è stato visualizzato.

[Ulteriori informazioni](#)

- BZPOPMIN

Rimuove e restituisce il membro con il punteggio più basso da uno o più set ordinati. Si blocca finché un membro non è disponibile. Elimina il set ordinato se l'ultimo elemento è stato visualizzato.

[Ulteriori informazioni](#)

- ZADD

Aggiunge uno o più membri a un set ordinato oppure ne aggiorna i punteggi. Crea la chiave, se non esiste.

[Ulteriori informazioni](#)

- ZCARD

Restituisce il numero di membri di un set ordinato.

[Ulteriori informazioni](#)

- ZCOUNT

Restituisce il numero di membri di un set ordinato con il punteggio compreso in un intervallo.

[Ulteriori informazioni](#)

- ZDIFF

Restituisce la differenza tra più set ordinati.

[Ulteriori informazioni](#)

- ZDIFFSTORE

Archivia la differenza di più set ordinati in una chiave.

[Ulteriori informazioni](#)

- ZINCRBY

Incrementa il punteggio di un membro in un set ordinato.

[Ulteriori informazioni](#)

- ZINTER

Restituisce l'intersezione di più set ordinati.

[Ulteriori informazioni](#)

- ZINTERCARD

Restituisce il numero di membri dell'intersezione di più set ordinati.

[Ulteriori informazioni](#)

- ZINTERSTORE

Archivia l'intersezione di più set ordinati in una chiave.

[Ulteriori informazioni](#)

- ZLEXCOUNT

Restituisce il numero di membri di un set ordinato in un intervallo lessicografico.

[Ulteriori informazioni](#)

- ZMPOP

Restituisce i membri con il punteggio più alto o più basso di uno o più set ordinati dopo averli rimossi. Elimina il set ordinato se l'ultimo membro è stato visualizzato.

[Ulteriori informazioni](#)

- ZMSCORE

Restituisce il punteggio di uno o più membri di un set ordinato.

[Ulteriori informazioni](#)

- ZPOPMAX

Restituisce i membri con il punteggio più alto di un set ordinato dopo averli rimossi. Elimina il set ordinato se l'ultimo membro è stato visualizzato.

[Ulteriori informazioni](#)

- ZPOPMIN

Restituisce i membri con il punteggio più basso di un set ordinato dopo averli rimossi. Elimina il set ordinato se l'ultimo membro è stato visualizzato.

[Ulteriori informazioni](#)

- ZRANDMEMBER

Restituisce uno o più membri casuali di un set ordinato.

[Ulteriori informazioni](#)

- ZRANGE

Restituisce i membri di un set ordinato in un intervallo di indici.

[Ulteriori informazioni](#)

- ZRANGEBYLEX

Restituisce i membri di un set ordinato in un intervallo lessicografico.

[Ulteriori informazioni](#)

- ZRANGEBYSCORE

Restituisce i membri di un set ordinato in un intervallo di punteggi.

[Ulteriori informazioni](#)

- ZRANGESTORE

Archivia in una chiave un intervallo di membri di un set ordinato.

[Ulteriori informazioni](#)

- ZRANK

Restituisce l'indice di un membro in un set ordinato per punteggi in ordine crescente.

[Ulteriori informazioni](#)

- ZREM

Rimuove uno o più membri da un set ordinato. Elimina il set ordinato se tutti i membri sono stati rimossi.

[Ulteriori informazioni](#)

- ZREMRANGEBYLEX

Rimuove i membri di un set ordinato in un intervallo lessicografico. Elimina il set ordinato se tutti i membri sono stati rimossi.

[Ulteriori informazioni](#)

- ZREMRANGEBYRANK

Rimuove i membri di un set ordinato in un intervallo di indici. Elimina il set ordinato se tutti i membri sono stati rimossi.

[Ulteriori informazioni](#)

- ZREMRANGEBYSCORE

Rimuove i membri di un set ordinato in un intervallo di punteggi. Elimina il set ordinato se tutti i membri sono stati rimossi.

[Ulteriori informazioni](#)

- ZREVRANGE

Restituisce i membri di un set ordinato in un intervallo di indici in ordine inverso.

[Ulteriori informazioni](#)

- ZREVRANGEBYLEX

Restituisce i membri di un set ordinato in un intervallo lessicografico in ordine inverso.

[Ulteriori informazioni](#)

- ZREVRANGEBYSCORE

Restituisce i membri di un set ordinato in un intervallo di punteggi in ordine inverso.

[Ulteriori informazioni](#)

- ZREVRANK

Restituisce l'indice di un membro in un set ordinato per punteggi in ordine decrescente.

[Ulteriori informazioni](#)

- ZSCAN

Esegue l'iterazione dei membri e dei punteggi di un set ordinato.

[Ulteriori informazioni](#)

- ZSCORE

Restituisce il punteggio di un membro in un set ordinato.

[Ulteriori informazioni](#)

- ZUNION

Restituisce l'unione di più set ordinati.

[Ulteriori informazioni](#)

- ZUNIONSTORE

Archivia l'unione di più set ordinati in una chiave.

[Ulteriori informazioni](#)

Comandi di flusso

- XACK

Restituisce il numero di messaggi confermati dal membro del gruppo di consumer di un flusso.

[Ulteriori informazioni](#)

- XADD

Aggiunge un nuovo messaggio a un flusso. Crea la chiave, se non esiste.

[Ulteriori informazioni](#)

- XAUTOCLAIM

Cambia o acquisisce la proprietà dei messaggi in un gruppo di consumer, come se i messaggi fossero recapitati a un membro del gruppo di consumer.

[Ulteriori informazioni](#)

- XCLAIM

Cambia o acquisisce la proprietà di un messaggio in un gruppo di consumer, come se il messaggio fosse stato recapitato a un membro del gruppo di consumer.

[Ulteriori informazioni](#)

- XDEL

Restituisce il numero di messaggi dopo averli rimossi da un flusso.

[Ulteriori informazioni](#)

- XGROUP CREATE

Crea un gruppo di consumer.

[Ulteriori informazioni](#)

- XGROUP CREATECONSUMER

Crea un consumer in un gruppo di consumer.

[Ulteriori informazioni](#)

- XGROUP DELCONSUMER

Elimina un consumer da un gruppo di consumer.

[Ulteriori informazioni](#)

- XGROUP DESTROY

Distrugge un gruppo di consumer.

[Ulteriori informazioni](#)

- XGROUP SETID

Imposta l'ultimo ID fornito di un gruppo di consumer.

[Ulteriori informazioni](#)

- XINFO CONSUMERS

Restituisce l'elenco dei consumer di un gruppo di consumer.

[Ulteriori informazioni](#)

- XINFO GROUPS

Restituisce l'elenco dei gruppi di consumer di un flusso.

[Ulteriori informazioni](#)

- XINFO STREAM

Restituisce le informazioni su un flusso.

[Ulteriori informazioni](#)

- XLEN

Restituisce il numero dei messaggi in un flusso.

[Ulteriori informazioni](#)

- XPENDING

Restituisce le informazioni e le voci dell'elenco delle voci in sospeso di un gruppo di consumer di un flusso.

[Ulteriori informazioni](#)

- XRANGE

Restituisce i messaggi di un flusso in un intervallo di ID.

[Ulteriori informazioni](#)

- XREAD

Restituisce i messaggi di più flussi con ID superiori a quelli richiesti. Si blocca finché un messaggio non è disponibile.

[Ulteriori informazioni](#)

- XREADGROUP

Restituisce i messaggi nuovi o storici di un flusso per un consumer di un gruppo. Si blocca finché un messaggio non è disponibile.

[Ulteriori informazioni](#)

- XREVRANGE

Restituisce i messaggi di un flusso in un intervallo di ID in ordine inverso.

[Ulteriori informazioni](#)

- XTRIM

Elimina i messaggi dall'inizio di un flusso.

[Ulteriori informazioni](#)

Comandi di stringa

- APPEND

Aggiunge una stringa al valore di una chiave. Crea la chiave, se non esiste.

[Ulteriori informazioni](#)

- DECR

Diminuisce di uno il valore intero di una chiave. Utilizza 0 come valore iniziale se la chiave non esiste.

[Ulteriori informazioni](#)

- DECRBY

Diminuisce di un numero il valore intero di una chiave. Utilizza 0 come valore iniziale se la chiave non esiste.

[Ulteriori informazioni](#)

- GET

Restituisce il valore di stringa di una chiave.

[Ulteriori informazioni](#)

- GETDEL

Restituisce il valore di stringa di una chiave dopo l'eliminazione della chiave.

[Ulteriori informazioni](#)

- GETEX

Restituisce il valore di stringa di una chiave dopo averne impostato l'ora di scadenza.

[Ulteriori informazioni](#)

- GETRANGE

Restituisce una sottostringa della stringa archiviata in una chiave.

[Ulteriori informazioni](#)

- GETSET

Restituisce il valore di stringa precedente di una chiave dopo averla impostata su un nuovo valore.

[Ulteriori informazioni](#)

- INCR

Incrementa di uno il valore intero di una chiave. Utilizza 0 come valore iniziale se la chiave non esiste.

[Ulteriori informazioni](#)

- INCRBY

Incrementa di un numero il valore intero di una chiave. Utilizza 0 come valore iniziale se la chiave non esiste.

[Ulteriori informazioni](#)

- INCRBYFLOAT

Incrementa di un numero il valore in virgola mobile di una chiave. Utilizza 0 come valore iniziale se la chiave non esiste.

[Ulteriori informazioni](#)

- LCS

Trova la sottostringa comune più lunga.

[Ulteriori informazioni](#)

- MGET

Restituisce atomicamente i valori di stringa di una o più chiavi.

[Ulteriori informazioni](#)

- MSET

Crea o modifica atomicamente i valori di stringa di una o più chiavi.

[Ulteriori informazioni](#)

- MSETNX

Modifica atomicamente i valori di stringa di una o più chiavi solo quando non esistono tutte le chiavi.

[Ulteriori informazioni](#)

- PSETEX

Imposta il valore di stringa e l'ora di scadenza in millisecondi di una chiave. La chiave viene creata se non esiste.

[Ulteriori informazioni](#)

- SET

Imposta il valore di stringa di una chiave, ignorandone il tipo. La chiave viene creata se non esiste.

[Ulteriori informazioni](#)

- SETEX

Imposta il valore di stringa e l'ora di scadenza di una chiave. Crea la chiave, se non esiste.

[Ulteriori informazioni](#)

- SETNX

Imposta il valore di stringa di una chiave solo quando la chiave non esiste.

[Ulteriori informazioni](#)

- SETRANGE

Sovrascrive una parte del valore di stringa mediante un offset. Crea la chiave, se non esiste.

[Ulteriori informazioni](#)

- STRLEN

Restituisce la lunghezza di un valore di stringa.

[Ulteriori informazioni](#)

- SUBSTR

Restituisce una sottostringa da un valore di stringa.

[Ulteriori informazioni](#)

Comandi di transazione

- DISCARD

Elimina una transazione.

[Ulteriori informazioni](#)

- EXEC

Esegue tutti i comandi in una transazione.

[Ulteriori informazioni](#)

- MULTI

Inizia una transazione.

[Ulteriori informazioni](#)

Comandi Redis limitati

Per fornire un'esperienza di servizio gestito, ElastiCache limita l'accesso a comandi specifici del motore di cache che richiedono privilegi avanzati. Per le cache che eseguono Redis, i seguenti comandi non sono disponibili:

- `acl setuser`
- `acl load`
- `acl save`
- `acl deluser`
- `bgrewriteaof`
- `bgsave`

- `cluster addslot`
- `cluster addslotsrange`
- `cluster bumpepoch`
- `cluster delslot`
- `cluster delslotsrange`
- `cluster failover`
- `cluster flushslots`
- `cluster forget`
- `cluster links`
- `cluster meet`
- `cluster setslot`
- `config`
- `debug`
- `migrate`
- `psync`
- `replicaof`
- `save`
- `slaveof`
- `shutdown`
- `sync`

Inoltre, i seguenti comandi non sono disponibili per le cache serverless:

- `acl log`
- `client caching`
- `client getredir`
- `client id`
- `client info`
- `client kill`
- `client list`

- `client no-evict`
- `client pause`
- `client tracking`
- `client trackinginfo`
- `client unblock`
- `client unpause`
- `cluster count-failure-reports`
- `fcall`
- `fcall_ro`
- `function`
- `function delete`
- `function dump`
- `function flush`
- `function help`
- `function kill`
- `function list`
- `function load`
- `function restore`
- `function stats`
- `keys`
- `lastsave`
- `latency`
- `latency doctor`
- `latency graph`
- `latency help`
- `latency histogram`
- `latency history`
- `latency latest`
- `latency reset`

- `memory`
- `memory doctor`
- `memory help`
- `memory malloc-stats`
- `memory purge`
- `memory stats`
- `memory usage`
- `monitor`
- `move`
- `object`
- `object encoding`
- `object freq`
- `object help`
- `object idletime`
- `object refcount`
- `pfdebug`
- `pfselftest`
- `psubscribe`
- `pubsub numpat`
- `punsubscribe`
- `script kill`
- `slowlog`
- `slowlog get`
- `slowlog help`
- `slowlog len`
- `slowlog reset`
- `swapdb`
- `unwatch`
- `wait`
- `watch`

Configurazione e limiti di Redis

Il motore Redis fornisce una serie di parametri di configurazione, alcuni dei quali sono modificabili in ElastiCache per Redis, mentre altri non sono modificabili al fine di fornire prestazioni e affidabilità stabili.

Cache serverless

Per le cache serverless, i gruppi di parametri non sono utilizzati e tutta la configurazione Redis non è modificabile. Sono forniti i seguenti parametri Redis:

Nome	Informazioni	Descrizione
<code>acl-pubsub-default</code>	<code>allchannels</code>	Le autorizzazioni predefinite del canale pubsub per gli utenti ACL nella cache.
<code>client-output-buffer-limit</code>	<code>normal 0 0 0</code> <code>pubsub 32mb 8mb 60</code>	I client normali non hanno limiti di buffer. I client PUB/SUB vengono disconnessi se violano il backlog di 32 MiB o violano il backlog di 8 MiB per 60 secondi.
<code>client-query-buffer-limit</code>	1 GiB	La dimensione massima di un singolo buffer di query dei client. Inoltre, i client non possono generare una richiesta con più di 4.000 argomenti.
<code>cluster-allow-pubsubshard-when-down</code>	yes	In tal modo la cache può servire il traffico pubsub mentre è parzialmente inattiva.
<code>cluster-allow-reads-when-down</code>	yes	In tal modo la cache può servire il traffico di lettura mentre è parzialmente inattiva.
<code>cluster-enabled</code>	yes	Tutte le cache serverless sono abilitate alla modalità cluster, che consente loro di partizion

Nome	Informazioni	Descrizione
		are in modo trasparente i dati su più partizion i di back-end. Tutti gli slot vengono mostrati ai client come fossero di proprietà di un singolo nodo virtuale.
<code>cluster-require-full-coverage</code>	no	Quando il keyspace è parzialmente inattivo (ovvero almeno uno slot hash è inaccessibile), la cache continua ad accettare le query per la parte del keyspace ancora coperta. L'intero keyspace sarà sempre "coperto" da un singolo nodo virtuale in <code>cluster slots</code> .
<code>lua-time-limit</code>	5000	Il tempo di esecuzione massimo per uno script Lua, in millisecondi, prima che ElastiCache agisca in modo da interrompere lo script. Se <code>lua-time-limit</code> viene superato, tutti i comandi Redis restituiscono un errore nel formato <code>____-BUSY</code> . Poiché questo stato può causare interferenza con molte operazioni Redis essenziali, ElastiCache emetterà prima un comando <code>SCRIPT KILL</code> . In caso di esito negativo, ElastiCache forzerà il riavvio di Redis.
<code>maxclients</code>	65000	Il numero massimo di client che possono essere connessi contemporaneamente alla cache. Le altre connessioni possono essere o meno completate.
<code>maxmemory-policy</code>	<code>volatile-lru</code>	Gli elementi con un valore TTL impostato vengono sottoposti a espulsione in base alla stima di quelli utilizzati meno di recente (LRU) quando viene raggiunto il limite di memoria della cache.

Nome	Informazioni	Descrizione
notify-keyspace-events	Stringa vuota	Gli eventi del keyspace non sono attualmente supportati nelle cache serverless.
port	Porta principale: 6379 Porta di lettura: 6380	Le cache serverless pubblicizzano due porte con lo stesso nome host. La porta principale consente scritture e letture, mentre la porta di lettura consente le letture a coerenza finale con una latenza inferiore utilizzando il comando READONLY.
proto-max-bulk-len	512 MiB	La dimensione massima di una singola richiesta di elementi.
timeout	0	La disconnessione dei client non viene forzata per un determinato periodo di inattività, ma può essere eseguita durante lo stato stazionario a beneficio del sistema di bilanciamento del carico.

Inoltre, sono applicati i limiti seguenti:

Nome	Informazioni	Descrizione
Lunghezza del nome della chiave	4 KiB	La dimensione massima del nome di una chiave Redis o un canale. I client che fanno riferimento a chiavi di dimensioni superiori restituiscono un errore.
Dimensione dello script Lua	4 MiB	La dimensione massima di un singolo script Lua in Redis. I tentativi di caricare uno script Lua più grande restituiscono un errore.

Nome	Informazioni	Descrizione
Dimensione dello slot	32 GiB	La dimensione massima di un singolo slot hash Redis. I client che cercano di impostare più dati su un singolo slot Redis attivano la policy di espulsione sullo slot e, se nessuna chiave può essere soggetta a espulsione, restituiscono un errore di memoria insufficiente (OOM).

Cluster progettati autonomamente

Per i cluster progettati autonomamente, consulta [Parametri specifici di Redis](#) per ottenere l'elenco dei valori predefiniti dei parametri di configurazione e per sapere quali sono configurabili. I valori predefiniti sono generalmente consigliati a meno che non si abbia un caso d'uso specifico che ne richieda la sostituzione.

Best practice con i client Redis

Scopri le best practice per gli scenari più comuni e segui gli esempi di codice di alcune delle più popolari librerie client Redis open source (redis-py, PHPRedis e Lettuce).

Argomenti

- [Numero elevato di connessioni](#)
- [Individuazione e backoff esponenziale dei client del cluster Redis](#)
- [Configurazione del timeout lato client](#)
- [Configurazione di un timeout di inattività sul lato server](#)
- [Script Lua in Redis](#)
- [Archiviazione di elementi composti di grandi dimensioni](#)
- [Configurazione del client Lettuce](#)
- [Esempi di client IPv6](#)

Numero elevato di connessioni

Le cache serverless e i singoli nodi ElastiCache per Redis supportano fino a 65.000 connessioni client simultanee. Tuttavia, per ottimizzare le prestazioni, consigliamo che le applicazioni client non

operino costantemente a un livello così elevato. Redis è un processo a thread singolo basato su un ciclo di eventi in cui le richieste dei client in entrata vengono gestite in sequenza. Ciò significa che il tempo di risposta di un determinato client si allunga con l'aumento del numero di client connessi.

Puoi intraprendere le seguenti azioni per evitare di imbatterti in un collo di bottiglia di connessione sul server Redis:

- Esegui le operazioni di lettura dalle repliche di lettura. Per farlo, utilizza gli endpoint di lettura ElastiCache in modalità cluster disabilitata o le repliche di lettura in modalità cluster abilitata, inclusa una cache serverless.
- Distribuisci il traffico di scrittura su più nodi primari. Sono disponibili due modi per farlo: puoi utilizzare un cluster Redis con più partizioni con un client compatibile con la modalità cluster Redis oppure puoi scrivere su più nodi primari in modalità cluster disabilitata con il partizionamento lato client. Questa operazione viene eseguita automaticamente in una cache serverless.
- Usa un pool di connessioni quando disponibile nella libreria client.

In generale, la creazione di una connessione TCP è un'operazione costosa a livello di calcolo rispetto ai comandi Redis tipici. Ad esempio, la gestione di una richiesta SET/GET viene eseguita a una velocità superiore quando si riutilizza una connessione esistente. L'uso di un pool di connessioni client di dimensioni limitate riduce il sovraccarico della gestione delle connessioni. Inoltre, limita il numero di connessioni simultanee in entrata dell'applicazione client.

Il seguente esempio di codice di PHPRedis mostra che viene creata una nuova connessione per ogni nuova richiesta utente:

```
$redis = new Redis();
if ($redis->connect($HOST, $PORT) != TRUE) {
    //ERROR: connection failed
    return;
}
$redis->set($key, $value);
unset($redis);
$redis = NULL;
```

Abbiamo confrontato questo codice in un ciclo su un'istanza Amazon Elastic Compute Cloud (Amazon EC2) connessa a un nodo Graviton2 (m6g.2xlarge) di ElastiCache per Redis. Abbiamo posizionato il client e il server nella stessa zona di disponibilità. La latenza media dell'intera operazione è stata di 2,82 millisecondi.

Quando abbiamo aggiornato il codice e utilizzato le connessioni persistenti e un pool di connessioni, la latenza media dell'intera operazione è stata di 0,21 millisecondi:

```
$redis = new Redis();
if ($redis->pconnect($HOST, $PORT) != TRUE) {
    // ERROR: connection failed
    return;
}
$redis->set($key, $value);
unset($redis);
$redis = NULL;
```

Configurazioni di redis.ini richieste:

- redis.pconnect.pooling_enabled=1
- redis.pconnect.connection_limit=10

Il codice seguente è un esempio di [pool di connessioni Redis-py](#):

```
conn = Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
    max_connections=10))
conn.set(key, value)
```

Il codice seguente è un esempio di [pool di connessioni Lettuce](#):

```
RedisClient client = RedisClient.create(RedisURI.create(HOST, PORT));
GenericObjectPool<StatefulRedisConnection> pool =
    ConnectionPoolSupport.createGenericObjectPool(() -> client.connect(), new
    GenericObjectPoolConfig());
pool.setMaxTotal(10); // Configure max connections to 10
try (StatefulRedisConnection connection = pool.borrowObject()) {
    RedisCommands syncCommands = connection.sync();
    syncCommands.set(key, value);
}
```

Individuazione e backoff esponenziale dei client del cluster Redis

Quando ci si connette a un cluster ElastiCache per Redis in modalità cluster abilitata, la libreria client Redis corrispondente deve riconoscere il cluster. I client devono ottenere una mappa degli slot

hash ai nodi corrispondenti del cluster per inviare le richieste ai nodi giusti ed evitare il sovraccarico delle prestazioni dovuto alla gestione dei reindirizzamenti del cluster. Di conseguenza, il client deve ottenere l'elenco completo degli slot e dei nodi mappati in due diverse situazioni:

- Il client è inizializzato e deve compilare la configurazione iniziale degli slot.
- Il server riceve un reindirizzamento MOVED, ad esempio in caso di failover in cui tutti gli slot serviti dal precedente nodo primario vengono acquisiti dalla replica oppure in caso di ripartizionamento quando gli slot vengono spostati dal nodo primario di origine al nodo primario di destinazione.

L'individuazione del client viene in genere effettuata tramite l'emissione di un comando CLUSTER SLOT o CLUSTER NODE al server Redis. Consigliamo il metodo CLUSTER SLOT perché restituisce al client il set di intervalli di slot e i nodi primari e di replica associati. Questo metodo non richiede un'analisi aggiuntiva da parte del client ed è più efficiente.

A seconda della topologia del cluster, la dimensione della risposta per il comando CLUSTER SLOT può variare in base alla dimensione del cluster. I cluster più grandi con più nodi producono una risposta di dimensione maggiore. Di conseguenza, è importante garantire che il numero di client che eseguono l'individuazione della topologia del cluster non cresca in modo illimitato. Ad esempio, quando l'applicazione client avvia o perde la connessione del server e deve eseguire l'individuazione del cluster, un errore comune è che l'applicazione client generi diverse richieste di riconnessione e individuazione senza aggiungere un backoff esponenziale al momento del nuovo tentativo. Ciò può impedire al server Redis di rispondere per un periodo di tempo prolungato, con un utilizzo della CPU al 100%. L'interruzione si prolunga se ogni comando CLUSTER SLOT deve elaborare un numero elevato di nodi nel bus del cluster. In passato abbiamo osservato diverse interruzioni dei client a causa di questo comportamento in diversi linguaggi, tra cui Python (redis-py-cluster) e Java (Lettuce e Redisson).

In una cache serverless, molti problemi vengono mitigati automaticamente perché la topologia del cluster pubblicizzata è statica ed è composta da due elementi: un endpoint di scrittura e un endpoint di lettura. L'individuazione del cluster viene inoltre distribuita automaticamente su più nodi quando si utilizza l'endpoint di cache. I seguenti suggerimenti sono tuttavia utili.

Per mitigare l'impatto causato da un afflusso improvviso di richieste di connessione e individuazione, ti consigliamo di procedere come segue:

- Implementa un pool di connessioni client di dimensioni limitate per contenere il numero di connessioni simultanee in entrata dell'applicazione client.

- Quando il client si disconnette dal server a causa di un timeout, riprova con un backoff esponenziale con jitter. In tal modo eviti che più client sovraccarichino il server contemporaneamente.
- Utilizza le indicazioni contenute in [Individuazione degli endpoint di connessione](#) per trovare l'endpoint del cluster ed eseguire l'individuazione del cluster. In questo modo, distribuisce il carico dell'individuazione su tutti i nodi del cluster (fino a 90) invece di usare i pochi nodi di inizializzazione codificati nel cluster.

Di seguito sono riportati alcuni esempi di codice per la logica di ripetizione dei tentativi del backoff esponenziale in redis-py, PHPRedis e Lettuce.

Esempio 1 di logica di backoff: redis-py

redis-py ha un meccanismo di ripetizione dei tentativi integrato che ripete il tentativo una volta subito dopo l'errore. Questo meccanismo può essere abilitato con l'argomento `retry_on_timeout` fornito durante la creazione di un oggetto [Redis](#). Di seguito è illustrato un meccanismo di ripetizione dei tentativi personalizzato con backoff esponenziale e jitter. Inviama una richiesta pull per implementare in modo nativo il backoff esponenziale in [redis-py \(#1494\)](#). In futuro potrebbe non essere necessario implementarlo manualmente.

```
def run_with_backoff(function, retries=5):
    base_backoff = 0.1 # base 100ms backoff
    max_backoff = 10 # sleep for maximum 10 seconds
    tries = 0
    while True:
        try:
            return function()
        except (ConnectionError, TimeoutError):
            if tries >= retries:
                raise
            backoff = min(max_backoff, base_backoff * (pow(2, tries) + random.random()))
            print(f"sleeping for {backoff:.2f}s")
            sleep(backoff)
            tries += 1
```

Quindi puoi utilizzare il codice seguente per impostare un valore:

```
client = redis.Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
    max_connections=10))
res = run_with_backoff(lambda: client.set("key", "value"))
```

```
print(res)
```

A seconda del carico di lavoro, è possibile modificare il valore di backoff di base da 1 secondo a poche decine o centinaia di millisecondi per i carichi di lavoro sensibili alla latenza.

Esempio 2 di logica di backoff: PHPRedis

PHPRedis ha un meccanismo di ripetizione dei tentativi integrato che ripete il tentativo per un massimo di 10 volte (non configurabile). Si verifica un ritardo configurabile tra i tentativi (con un jitter dal secondo tentativo in poi). Per ulteriori informazioni, consulta il seguente [codice di esempio](#). Inviando una richiesta pull per implementare in modo nativo il backoff esponenziale in [PHPRedis \(#1986\)](#) che è stato unito e [documentato](#). Per chi utilizza l'ultima versione di PHPRedis, non è necessario implementarlo manualmente, il riferimento incluso qui è per coloro che usano le versioni precedenti. Il seguente è un esempio di codice che configura il ritardo del meccanismo di ripetizione dei tentativi:

```
$timeout = 0.1; // 100 millisecond connection timeout
$retry_interval = 100; // 100 millisecond retry interval
$client = new Redis();
if($client->pconnect($HOST, $PORT, $timeout, NULL, $retry_interval) != TRUE) {
    return; // ERROR: connection failed
}
$client->set($key, $value);
```

Esempio 3 di logica di backoff: Lettuce

Lettuce dispone di meccanismi di ripetizione dei tentativi incorporati basati sulle strategie di backoff esponenziale descritte nel post [Exponential Backoff and Jitter](#). Di seguito è riportato un estratto di codice che mostra l'approccio completo del jitter:

```
public static void main(String[] args)
{
    ClientResources resources = null;
    RedisClient client = null;

    try {
        resources = DefaultClientResources.builder()
            .reconnectDelay(Delay.fullJitter(
                Duration.ofMillis(100), // minimum 100 millisecond delay
                Duration.ofSeconds(5), // maximum 5 second delay
            ))
            .build();
        client = new RedisClient(resources);
    } catch (Exception e) {
        // Handle exception
    }
}
```

```
    100, TimeUnit.MILLISECONDS) // 100 millisecond base
).build();

client = RedisClient.create(resources, RedisURI.create(HOST, PORT));
client.setOptions(ClientOptions.builder()
.socketOptions(SocketOptions.builder().connectTimeout(Duration.ofMillis(100)).build()) //
100 millisecond connection timeout
.timeoutOptions(TimeoutOptions.builder().fixedTimeout(Duration.ofSeconds(5)).build()) //
5 second command timeout
.build());

    // use the connection pool from above example
} finally {
    if (connection != null) {
        connection.close();
    }

    if (client != null){
        client.shutdown();
    }

    if (resources != null){
        resources.shutdown();
    }

}
}
```

Configurazione del timeout lato client

Configura il timeout lato client in modo appropriato per assegnare al server il tempo sufficiente per elaborare la richiesta e generare la risposta. Ciò consente inoltre di anticipare l'errore (fail fast) se non è possibile stabilire la connessione al server. Alcuni comandi Redis possono comportare costi di calcolo più elevati rispetto ad altri. Ad esempio, gli script Lua o le transazioni MULTI/EXEC che contengono più comandi che devono essere eseguiti atomicamente. In generale, si consiglia un timeout lato client più elevato per evitare il timeout del client prima che la risposta venga ricevuta dal server, tra cui i seguenti scenari:

- Esecuzione di comandi su più chiavi
- Esecuzione di transazioni MULTI/EXEC o script Lua costituiti da più comandi Redis
- Lettura di valori di grandi dimensioni

- Esecuzione di operazioni di blocco come BLPOP

Nel caso di un'operazione di blocco come BLPOP, la best practice consiste nell'impostare il timeout del comando su un valore inferiore al timeout del socket.

Di seguito sono riportati esempi di codice per l'implementazione di un timeout lato client in redis-py, PHPRedis e Lettuce.

Esempio 1 di configurazione del timeout: redis-py

Di seguito è riportato un esempio di codice con redis-py:

```
# connect to Redis server with a 100 millisecond timeout
# give every Redis command a 2 second timeout
client = redis.Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
    max_connections=10,socket_connect_timeout=0.1,socket_timeout=2))

res = client.set("key", "value") # will timeout after 2 seconds
print(res)                       # if there is a connection error

res = client.blpop("list", timeout=1) # will timeout after 1 second
                                     # less than the 2 second socket timeout
print(res)
```

Esempio 2 di configurazione del timeout: PHPRedis

Di seguito è riportato un esempio di codice con PHPRedis:

```
// connect to Redis server with a 100ms timeout
// give every Redis command a 2s timeout
$client = new Redis();
$timeout = 0.1; // 100 millisecond connection timeout
$retry_interval = 100; // 100 millisecond retry interval
$client = new Redis();
if($client->pconnect($HOST, $PORT, 0.1, NULL, 100, $read_timeout=2) != TRUE){
    return; // ERROR: connection failed
}
$client->set($key, $value);

$res = $client->set("key", "value"); // will timeout after 2 seconds
print "$res\n";                    // if there is a connection error
```

```
$res = $client->blpop("list", 1); // will timeout after 1 second
print "$res\n";                // less than the 2 second socket timeout
```

Esempio 3 di configurazione del timeout: Lettuce

Di seguito è riportato un esempio di codice con Lettuce:

```
// connect to Redis server and give every command a 2 second timeout
public static void main(String[] args)
{
    RedisClient client = null;
    StatefulRedisConnection<String, String> connection = null;
    try {
        client = RedisClient.create(RedisURI.create(HOST, PORT));
        client.setOptions(ClientOptions.builder()
            .socketOptions(SocketOptions.builder().connectTimeout(Duration.ofMillis(100)).build()) //
            100 millisecond connection timeout
            .timeoutOptions(TimeoutOptions.builder().fixedTimeout(Duration.ofSeconds(2)).build()) //
            2 second command timeout
            .build());

        // use the connection pool from above example

        commands.set("key", "value"); // will timeout after 2 seconds
        commands.blpop(1, "list"); // BLPPOP with 1 second timeout
    } finally {
        if (connection != null) {
            connection.close();
        }

        if (client != null){
            client.shutdown();
        }
    }
}
```

Configurazione di un timeout di inattività sul lato server

Abbiamo osservato casi in cui l'applicazione di un cliente ha un numero elevato di client inattivi collegati, ma non invia attivamente comandi. In tali scenari, è possibile esaurire tutte le 65.000 connessioni con un numero elevato di client inattivi. Per evitare tali situazioni, configura l'impostazione del timeout in modo appropriato sul server con i [Parametri specifici di Redis](#). In tal

modo si garantisce che il server disconnetta attivamente i client inattivi per evitare un aumento del numero di connessioni. Questa impostazione non è disponibile nelle cache serverless.

Script Lua in Redis

Redis supporta più di 200 comandi, inclusi quelli per eseguire gli script Lua. Tuttavia, quando si tratta di script Lua, ci sono diverse situazioni che possono influire sulla memoria e sulla disponibilità di Redis.

Script Lua non parametrizzati

Ogni script Lua viene memorizzato nella cache del server Redis prima di essere eseguito. Gli script Lua non parametrizzati sono univoci, il che può comportare l'archiviazione di un gran numero di script Lua e il consumo di molta memoria da parte del server Redis. Per mitigare questo problema, assicurati che tutti gli script Lua siano parametrizzati ed esegui regolarmente `SCRIPT FLUSH` per pulire gli script Lua memorizzati nella cache, se necessario.

Il codice di esempio seguente mostra come utilizzare gli script parametrizzati. Innanzitutto, abbiamo un esempio di approccio non parametrizzato che non è consigliato e produce tre diversi script Lua memorizzati nella cache:

```
eval "return redis.call('set','key1','1')" 0
eval "return redis.call('set','key2','2')" 0
eval "return redis.call('set','key3','3')" 0
```

Utilizza invece lo schema seguente per creare un singolo script in grado di accettare i parametri passati:

```
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key1 1
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key2 2
eval "return redis.call('set',KEYS[1],ARGV[1])" 1 key3 3
```

Script Lua a esecuzione prolungata

Gli script Lua possono eseguire più comandi atomicamente, pertanto il completamento può richiedere più tempo rispetto a un normale comando Redis. Se lo script Lua esegue esclusivamente operazioni di sola lettura, puoi interromperlo a metà. Tuttavia, non appena lo script Lua esegue un'operazione di scrittura, non puoi più interromperlo e deve essere eseguito fino al completamento. Uno script Lua a esecuzione prolungata che muta può far sì che il server Redis non risponda per molto tempo. Per

mitigare questo problema, evita gli script Lua a esecuzione prolungata e testali in un ambiente di produzione.

Script Lua con scritture invisibili

Esistono alcuni modi in cui uno script Lua può continuare a scrivere nuovi dati in Redis anche quando Redis supera il valore di `maxmemory`:

- Lo script inizia quando il server Redis è al di sotto di `maxmemory` e contiene più operazioni di scrittura.
- Il primo comando di scrittura dello script non consuma memoria (come `DEL`), le successive operazioni di scrittura consumano memoria.
- È possibile mitigare questo problema configurando una policy di espulsione adeguata in un server Redis diverso da `noeviction`. In tal modo Redis può eliminare gli oggetti e liberare memoria tra gli script Lua.

Archiviazione di elementi composti di grandi dimensioni

In alcuni scenari, un'applicazione può archiviare in Redis elementi composti di grandi dimensioni (come un set di dati hash da più GB). Questa non è una best practice consigliata perché spesso comporta problemi di prestazioni in Redis. Ad esempio, il client può eseguire un comando `HGETALL` per recuperare l'intera raccolta di hash da più GB. Questa esecuzione può generare una notevole pressione sulla memoria del server Redis che memorizza nel buffer di output del client l'elemento di grandi dimensioni. Inoltre, per la migrazione degli slot in modalità cluster, ElastiCache non migra gli slot che contengono elementi con dimensioni serializzate superiori a 256 MB.

Per risolvere i problemi relativi agli elementi di grandi dimensioni, tieni in considerazione i seguenti suggerimenti:

- Suddividi l'elemento composto di grandi dimensioni in più elementi più piccoli. Ad esempio, suddividi una raccolta di hash di grandi dimensioni in singoli campi chiave-valore con uno schema di nome della chiave che rifletta in modo appropriato la raccolta, ad esempio utilizzando un prefisso comune nel nome della chiave per identificare la raccolta di elementi. Se è necessario accedere atomicamente a più campi della stessa raccolta, è possibile utilizzare il comando `MGET` per recuperare più chiavi-valori nello stesso comando.
- Se hai valutato tutte le opzioni ma non riesci ancora a suddividere il set di dati della raccolta di grandi dimensioni, prova a utilizzare comandi che operano su un sottoinsieme dei dati della raccolta anziché sull'intera raccolta. Evita il caso d'uso che richiede il recupero atomico dell'intera

raccolta da più GB con lo stesso comando. Un esempio è l'utilizzo del comando HGET o HMGET anziché HGETALL nelle raccolte di hash.

Configurazione del client Lettuce

Questa sezione descrive le opzioni di configurazione consigliate di Java e Lettuce e come si applicano ai cluster ElastiCache.

I suggerimenti in questa sezione sono stati testati con Lettuce versione 6.2.2.

Argomenti

- [Esempio: configurazione Lettuce per la modalità cluster e TLS abilitato](#)
- [Esempio: configurazione Lettuce per la modalità cluster disabilitata e TLS abilitato](#)

TTL della cache DNS Java

Java Virtual Machine (JVM) memorizza nella cache le ricerche dei nomi DNS. Quando la JVM risolve un nome host in un indirizzo IP, memorizza nella cache l'indirizzo IP per un determinato periodo di tempo, noto come time-to-live (TTL).

La scelta di un valore TTL è un compromesso tra latenza e reattività al cambiamento. Con TTL più brevi, i resolver DNS notano gli aggiornamenti nel DNS del cluster più velocemente. Ciò può far sì che l'applicazione risponda più rapidamente alle sostituzioni o ad altri flussi di lavoro a cui è sottoposto il cluster. Tuttavia, se il TTL è troppo basso, aumenta il volume delle query, il che può aumentare la latenza dell'applicazione. Sebbene non esista un valore TTL corretto, quando si imposta il valore TTL è opportuno valutare il tempo che è possibile attendere affinché una modifica abbia effetto.

Poiché i nodi ElastiCache utilizzano voci del nome DNS che occasionalmente cambiano, consigliamo di configurare la JVM con un valore TTL compreso tra 5 e 10 secondi. In tal modo si garantisce che quando l'indirizzo IP di un nodo cambia, l'applicazione potrà ricevere e utilizzare il nuovo indirizzo IP della risorsa richiedendo la voce DNS.

In alcune configurazioni Java, il TTL predefinito di JVM è impostato in modo da non aggiornare mai le voci DNS finché JVM non viene riavviato.

Per dettagli su come impostare il TTL JVM, consulta [How to set the JVM TTL](#) (Come impostare il TTL JVM).

Versione Lettuce

Consigliamo la versione Lettuce 6.2.2 o versioni successive.

Endpoints

Quando si utilizzano cluster abilitati in modalità cluster, impostare `redisUri` sull'endpoint di configurazione del cluster. La ricerca DNS per questo URI restituisce un elenco di tutti i nodi disponibili nel cluster e viene risolta casualmente in uno di essi durante l'inizializzazione del cluster. Per ulteriori dettagli su come funziona l'aggiornamento della topologia, consulta `dynamicRefreshResources` più avanti in questo argomento.

SocketOption

Abilitazione di [KeepAlive](#). L'abilitazione di questa opzione riduce la necessità di gestire le connessioni non riuscite durante il runtime del comando.

Assicurarsi di impostare il [timeout di connessione](#) in base ai requisiti dell'applicazione e al carico di lavoro. Per ulteriori informazioni, consulta la sezione relativa ai timeout più avanti in questo argomento.

ClusterClientOption: opzioni client abilitate in modalità cluster

Abilitare [AutoReconnect](#) quando la connessione viene interrotta.

Impostazione di [CommandTimeout](#). Per ulteriori dettagli, consulta la sezione relativa ai timeout più avanti in questo argomento.

Impostare [NodeFilter](#) per filtrare i nodi non riusciti dalla topologia. Lettuce salva tutti i nodi che si trovano nell'output dei "nodi del cluster" (inclusi i nodi con stato PFAIL/FAIL) nelle "partizioni" del client (note anche come partizioni). Durante il processo di creazione della topologia del cluster, tenta di connettersi a tutti i nodi della partizione. Questo comportamento di Lettuce, che consiste nell'aggiungere nodi non riusciti, può causare errori di connessione (o avvisi) quando i nodi vengono sostituiti per qualsiasi motivo.

Ad esempio, al termine di un failover e quando il cluster avvia il processo di ripristino, mentre `clusterTopology` si aggiorna, la mappa dei nodi del bus del cluster prevede un breve periodo di tempo in cui il nodo inattivo viene elencato come nodo non riuscito (FAIL), prima che venga completamente rimosso dalla topologia. Durante questo periodo, il client Lettuce Redis lo considera un nodo integro e si connette continuamente ad esso. Ciò causa un errore dopo che il nuovo tentativo è esaurito.

Per esempio:

```
final ClusterClientOptions clusterClientOptions =
    ClusterClientOptions.builder()
        ... // other options
        .nodeFilter(it ->
            ! (it.is(RedisClusterNode.NodeFlag.FAIL)
                || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
                || it.is(RedisClusterNode.NodeFlag.HANDSHAKE)
                || it.is(RedisClusterNode.NodeFlag.NOADDR)))
        .validateClusterNodeMembership(false)
        .build();
redisClusterClient.setOptions(clusterClientOptions);
```

Note

Il filtraggio dei nodi viene utilizzato al meglio con `DynamicRefreshSources` impostato su `true`. Altrimenti, se la vista della topologia viene presa da un singolo nodo di origine problematico, che rileva un nodo primario non riuscito di qualche partizione, filtrerà questo nodo primario, il che comporterà la mancata copertura degli slot. La presenza di più nodi di origine (quando `DynamicRefreshSources` è `true`) riduce la probabilità che si verifichi questo problema, poiché almeno alcuni dei nodi di origine dovrebbero avere una vista topologica aggiornata dopo un failover con il primario appena promosso.

`ClusterTopologyRefreshOptions`: opzioni per il controllo dell'aggiornamento della topologia del cluster del client abilitato alla modalità cluster

Note

I cluster con la modalità cluster disabilitata non supportano i comandi di rilevamento del cluster e non sono compatibili con la funzionalità di rilevamento della topologia dinamica di tutti i client.

La modalità cluster disabilitata con ElastiCache non è compatibile con `MasterSlaveTopologyRefresh` di Lettuce. Invece, per la modalità cluster disabilitata è possibile configurare un `StaticMasterReplicaTopologyProvider` e fornire gli endpoint di lettura e scrittura del cluster.

Per ulteriori informazioni sulla connessione a cluster disattivati, consulta [Individuazione degli endpoint di un cluster Redis \(modalità cluster disabilitata\) \(console\)](#).

Se desideri utilizzare la funzionalità di rilevamento della topologia dinamica di Lettuce, puoi creare un cluster con la modalità cluster abilitata con la stessa configurazione di partizioni del cluster esistente. Tuttavia, per i cluster abilitati alla modalità cluster, consigliamo di configurare almeno 3 shard con almeno una replica per supportare il failover rapido.

Abilitazione di [enablePeriodicRefresh](#). Ciò consente aggiornamenti periodici della topologia del cluster in modo che il client aggiorni la topologia del cluster negli intervalli refreshPeriod (impostazione predefinita: 60 secondi). Quando è disabilitato, il client aggiorna la topologia del cluster solo quando si verificano errori quando tenta di eseguire comandi sul cluster.

Con questa opzione abilitata, è possibile ridurre la latenza associata all'aggiornamento della topologia del cluster aggiungendo questo processo a un'attività in background. Sebbene l'aggiornamento della topologia venga eseguito in un processo in background, può essere piuttosto lento per i cluster con molti nodi. Questo perché vengono eseguite query sulle viste di tutti i nodi per ottenere la vista del cluster più aggiornata. Se si gestisce un cluster di grandi dimensioni, si potrebbe voler aumentare il periodo.

Abilitazione di [enableAllAdaptiveRefreshTriggers](#). Ciò consente l'aggiornamento adattivo della topologia che utilizza tutti i [trigger](#): MOVED_REDIRECT, ASK_REDIRECT, PERSISTENT_RECONNECTS, UNCOVERED_SLOT, UNKNOWN_NODE. I trigger di aggiornamento adattivi avviano gli aggiornamenti della vista della topologia in base agli eventi che si verificano durante le operazioni del cluster Redis. L'attivazione di questa opzione comporta un aggiornamento immediato della topologia quando si verifica uno dei trigger precedenti. Gli aggiornamenti attivati adattivi hanno una frequenza limitata mediante un timeout perché gli eventi possono verificarsi su larga scala (timeout predefinito tra gli aggiornamenti: 30).

Abilitazione di [closeStaleConnections](#). Ciò consente di chiudere le connessioni obsolete durante l'aggiornamento della topologia del cluster. Ha effetto solo se [ClusterTopologyRefreshOptions.isPeriodicRefreshEnabled\(\)](#) è true. Quando è abilitato, il client può chiudere le connessioni obsolete e crearne di nuove in background. Ciò riduce la necessità di gestire le connessioni non riuscite durante il runtime del comando.

Abilitazione di [dynamicRefreshResources](#). Si consiglia di abilitare dynamicRefreshResources per i cluster di piccole dimensioni e di disabilitarlo per i cluster di grandi dimensioni. dynamicRefreshResources consente di rilevare i nodi del cluster dal nodo di origine fornito (ad esempio, l'endpoint di configurazione del cluster). Utilizza tutti i nodi rilevati come origine per aggiornare la topologia del cluster.

Utilizzando l'aggiornamento dinamico, esegue query su tutti i nodi rilevati per la topologia del cluster e tenta di scegliere la vista del cluster più accurata. Se è impostato su `false`, solo i nodi di origine iniziali vengono utilizzati come origini per l'individuazione della topologia e il numero di client viene ottenuto solo per i nodi di origine iniziali. Quando è disabilitato, se l'endpoint di configurazione del cluster viene risolto in un nodo non riuscito, il tentativo di aggiornare la vista del cluster restituisce un errore e genera delle eccezioni. Questo scenario può verificarsi perché è necessario del tempo prima che un nodo non riuscito venga rimosso dall'endpoint di configurazione del cluster. Pertanto, l'endpoint di configurazione può ancora essere risolto casualmente in un nodo non riuscito per un breve lasso di tempo.

Quando è abilitato, tuttavia, utilizziamo tutti i nodi del cluster ricevuti dalla vista del cluster per eseguire query sulla vista corrente. Poiché filtriamo i nodi non riusciti da quella vista, l'aggiornamento della topologia avverrà correttamente. Tuttavia, quando `dynamicRefreshSources` è `true`, Lettuce esegue query su tutti i nodi per ottenere la vista del cluster e quindi confronta i risultati. Pertanto, l'operazione può essere costosa per i cluster con molti nodi. Consigliamo di disattivare questa funzionalità per i cluster con molti nodi.

```
final ClusterTopologyRefreshOptions topologyOptions =
    ClusterTopologyRefreshOptions.builder()
        .enableAllAdaptiveRefreshTriggers()
        .enablePeriodicRefresh()
        .dynamicRefreshSources(true)
        .build();
```

ClientResources

Configurazione di [DnsResolver](#) con [DirContextDnsResolver](#). Il resolver DNS è basato su `com.sun.jndi.dns.DnsContextFactory` di Java.

Configurazione di [reconnectDelay](#) con backoff esponenziale e jitter completo. Lettuce ha meccanismi di ripetizione incorporati basati sulle strategie di backoff esponenziale. Per ulteriori informazioni, consulta [Exponential Backoff and Jitter](#) (Backoff esponenziale e Jitter) nel blog di architettura AWS (Architecture Blog). Per ulteriori informazioni sull'importanza di adottare una strategia di backoff dei nuovi tentativi, consulta le sezioni sulla logica di backoff del [post del blog sulle best practice](#) sul blog del database AWS.

```
ClientResources clientResources = DefaultClientResources.builder()
    .dnsResolver(new DirContextDnsResolver())
    .reconnectDelay(
        Delay.fullJitter(
```

```

        Duration.ofMillis(100),    // minimum 100 millisecond delay
        Duration.ofSeconds(10),    // maximum 10 second delay
        100, TimeUnit.MILLISECONDS)) // 100 millisecond base
    .build();

```

Timeout

Usa un valore di timeout di connessione inferiore rispetto al timeout dei comandi. Lettuce utilizza l'avvio lento della connessione. Quindi, se il timeout di connessione è superiore al timeout del comando, si può avere un periodo di errore persistente dopo un aggiornamento della topologia se Lettuce tenta di connettersi a un nodo non integro e il timeout del comando viene sempre superato.

Utilizzo di un timeout dinamico dei comandi per diversi comandi. Consigliamo di impostare il timeout dei comandi in base alla durata prevista dei comandi. Ad esempio, utilizzare un timeout più lungo per i comandi che iterano su più chiavi, come FLUSHDB, FLUSHALL, KEYS, SMEMBERS o gli script Lua. Utilizzare timeout più brevi per i comandi a chiave singola, come SET, GET e HSET.

Note

I timeout configurati nell'esempio seguente riguardano i test che eseguono comandi SET/GET con chiavi e valori lunghi fino a 20 byte. Il tempo di elaborazione può essere più lungo quando i comandi sono complessi o le chiavi e i valori sono più grandi. È necessario impostare i timeout in base al caso d'uso dell'applicazione.

```

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

```

```

SocketOptions socketOptions = SocketOptions.builder()
    .connectTimeout(CONNECT_TIMEOUT)
    .build();

```

```

class DynamicClusterTimeout extends TimeoutSource {
    private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
    ImmutableSet.<ProtocolKeyword>builder()
        .add(CommandType.FLUSHDB)
        .add(CommandType.FLUSHALL)
        .add(CommandType.CLUSTER)

```



```

        .add(CommandType.INFO)
        .add(CommandType.KEYS)
        .build();

private final Duration defaultCommandTimeout;
private final Duration metaCommandTimeout;

DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
{
    defaultCommandTimeout = defaultTimeout;
    metaCommandTimeout = metaTimeout;
}

@Override
public long getTimeout(RedisCommand<?, ?, ?> command) {
    if (META_COMMAND_TYPES.contains(command.getType())) {
        return metaCommandTimeout.toMillis();
    }
    return defaultCommandTimeout.toMillis();
}
}

// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
TimeoutOptions timeoutOptions = TimeoutOptions.builder()
    .timeoutSource(
        new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT, META_COMMAND_TIMEOUT))
    .build();

```

Esempio: configurazione Lettuce per la modalità cluster e TLS abilitato

Note

I timeout configurati nell'esempio seguente riguardano i test che eseguono comandi SET/GET con chiavi e valori lunghi fino a 20 byte. Il tempo di elaborazione può essere più lungo quando i comandi sono complessi o le chiavi e i valori sono più grandi. È necessario impostare i timeout in base al caso d'uso dell'applicazione.

```

// Set DNS cache TTL
public void setJVMProperties() {
    java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

```

```
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the cluster configuration endpoint
clusterConfigurationEndpoint = <cluster-configuration-endpoint> // TODO: add your
    cluster configuration endpoint
final RedisURI redisUriCluster =
    RedisURI.Builder.redis(clusterConfigurationEndpoint)
        .withPort(6379)
        .withSsl(true)
        .build();

// Configure the client's resources
ClientResources clientResources = DefaultClientResources.builder()
    .reconnectDelay(
        Delay.fullJitter(
            Duration.ofMillis(100),    // minimum 100 millisecond delay
            Duration.ofSeconds(10),    // maximum 10 second delay
            100, TimeUnit.MILLISECONDS)) // 100 millisecond base
    .dnsResolver(new DirContextDnsResolver())
    .build();

// Create a cluster client instance with the URI and resources
RedisClusterClient redisClusterClient =
    RedisClusterClient.create(clientResources, redisUriCluster);

// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
class DynamicClusterTimeout extends TimeoutSource {
    private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
    ImmutableSet.<ProtocolKeyword>builder()
        .add(CommandType.FLUSHDB)
        .add(CommandType.FLUSHALL)
        .add(CommandType.CLUSTER)
        .add(CommandType.INFO)
        .add(CommandType.KEYS)
        .build();

    private final Duration metaCommandTimeout;
    private final Duration defaultCommandTimeout;
```

```
DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
{
    defaultCommandTimeout = defaultTimeout;
    metaCommandTimeout = metaTimeout;
}

@Override
public long getTimeout(RedisCommand<?, ?, ?> command) {
    if (META_COMMAND_TYPES.contains(command.getType())) {
        return metaCommandTimeout.toMillis();
    }
    return defaultCommandTimeout.toMillis();
}
}

TimeoutOptions timeoutOptions = TimeoutOptions.builder()
    .timeoutSource(new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT,
META_COMMAND_TIMEOUT))
    .build();

// Configure the topology refreshment options
final ClusterTopologyRefreshOptions topologyOptions =
    ClusterTopologyRefreshOptions.builder()
        .enableAllAdaptiveRefreshTriggers()
        .enablePeriodicRefresh()
        .dynamicRefreshSources(true)
        .build();

// Configure the socket options
final SocketOptions socketOptions =
    SocketOptions.builder()
        .connectTimeout(CONNECT_TIMEOUT)
        .keepAlive(true)
        .build();

// Configure the client's options
final ClusterClientOptions clusterClientOptions =
    ClusterClientOptions.builder()
        .topologyRefreshOptions(topologyOptions)
        .socketOptions(socketOptions)
        .autoReconnect(true)
        .timeoutOptions(timeoutOptions)
        .nodeFilter(it ->
```

```

        ! (it.is(RedisClusterNode.NodeFlag.FAIL)
        || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
        || it.is(RedisClusterNode.NodeFlag.NOADDR)))
        .validateClusterNodeMembership(false)
        .build();

redisClusterClient.setOptions(clusterClientOptions);

// Get a connection
final StatefulRedisClusterConnection<String, String> connection =
    redisClusterClient.connect();

// Get cluster sync/async commands
RedisAdvancedClusterCommands<String, String> sync = connection.sync();
RedisAdvancedClusterAsyncCommands<String, String> async = connection.async();

```

Esempio: configurazione Lettuce per la modalità cluster disabilitata e TLS abilitato

Note

I timeout configurati nell'esempio seguente riguardano i test che eseguono comandi SET/GET con chiavi e valori lunghi fino a 20 byte. Il tempo di elaborazione può essere più lungo quando i comandi sono complessi o le chiavi e i valori sono più grandi. È necessario impostare i timeout in base al caso d'uso dell'applicazione.

```

// Set DNS cache TTL
public void setJVMProperties() {
    java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the primary/reader endpoint
clusterEndpoint = <primary/reader-endpoint> // TODO: add your node endpoint
RedisURI redisUriStandalone =

    RedisURI.Builder.redis(clusterEndpoint).withPort(6379).withSsl(true).withDatabase(0).build();

```

```
ClientResources clientResources =
    DefaultClientResources.builder()
        .dnsResolver(new DirContextDnsResolver())
        .reconnectDelay(
            Delay.fullJitter(
                Duration.ofMillis(100), // minimum 100 millisecond delay
                Duration.ofSeconds(10), // maximum 10 second delay
                100,
                TimeUnit.MILLISECONDS)) // 100 millisecond base
        .build();

// Use a dynamic timeout for commands, to avoid timeouts during
// slow operations.
class DynamicTimeout extends TimeoutSource {
    private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
    ImmutableSet.<ProtocolKeyword>builder()
        .add(CommandType.FLUSHDB)
        .add(CommandType.FLUSHALL)
        .add(CommandType.INFO)
        .add(CommandType.KEYS)
        .build();

    private final Duration metaCommandTimeout;
    private final Duration defaultCommandTimeout;

    DynamicTimeout(Duration defaultTimeout, Duration metaTimeout)
    {
        defaultCommandTimeout = defaultTimeout;
        metaCommandTimeout = metaTimeout;
    }

    @Override
    public long getTimeout(RedisCommand<?, ?, ?> command) {
        if (META_COMMAND_TYPES.contains(command.getType())) {
            return metaCommandTimeout.toMillis();
        }
        return defaultCommandTimeout.toMillis();
    }
}

TimeoutOptions timeoutOptions = TimeoutOptions.builder()
    .timeoutSource(new DynamicTimeout(DEFAULT_COMMAND_TIMEOUT, META_COMMAND_TIMEOUT))
    .build();
```

```
final SocketOptions socketOptions =
    SocketOptions.builder().connectTimeout(CONNECT_TIMEOUT).keepAlive(true).build();

ClientOptions clientOptions =

    ClientOptions.builder().timeoutOptions(timeoutOptions).socketOptions(socketOptions).build();

RedisClient redisClient = RedisClient.create(clientResources, redisUriStandalone);
redisClient.setOptions(clientOptions);
```

Esempi di client IPv6

Di seguito sono riportate le migliori pratiche per interagire con risorse compatibili con IPv6 con librerie client open source di uso comune. ElastiCache È possibile visualizzare [le best practice esistenti con cui interagire per](#) ottenere consigli sulla configurazione dei ElastiCache client per le risorse. ElastiCache Tuttavia, ci sono alcuni avvertenze che vale la pena segnalare quando si interagisce con risorse abilitate per IPv6.

Clients convalidati

ElastiCache è compatibile con Redis open source. Ciò significa che i client Redis open source che supportano le connessioni IPv6 dovrebbero essere in grado di connettersi a IPv6 abilitati per i cluster Redis. ElastiCache Inoltre, molti dei client Python e Java più diffusi sono stati collaudati e convalidati specificamente per funzionare con tutte le configurazioni di tipi di rete supportate (solo IPv4, solo IPv6 e dual-stack)

Clients convalidati:

- [Redis Py \(\) – 4,1.2](#)
- [Lettuce – Versione: 6.1.6.RELEASE](#)
- [Jedis – Versione: 3.6.0](#)

Configurazione di un protocollo preferito per i cluster dual-stack

Per cluster Redis (modalità cluster abilitata), puoi controllare il protocollo che verrà utilizzato dai client per connettersi ai nodi del cluster con il parametro IP Discovery. Il parametro IP Discovery può essere impostato su IPv4 o IPv6.

Per i cluster Redis, il parametro IP Discovery imposta il protocollo IP utilizzato nell'output [slot del cluster \(\)](#), [partizioni del cluster \(\)](#) e [nodi del cluster \(\)](#). Questi comandi vengono utilizzati dai client per

scoprire la topologia del cluster. I client utilizzano gli IP in questi comandi per connettersi agli altri nodi nel cluster.

La modifica di IP Discovery non comporterà alcun tempo di inattività per i client connessi. Tuttavia, la propagazione delle modifiche richiederà tempo. Per determinare quando la propagazione delle modifiche è terminata per un cluster Redis, monitorare l'output di `cluster slots`. Dopo che tutti i nodi restituiti dal comando `slot` del cluster segnalano IP con il nuovo protocollo, la propagazione delle modifiche è terminata.

Esempio con Redis-Py:

```
cluster = RedisCluster(host="xxxx", port=6379)
target_type = IPv6Address # Or IPv4Address if changing to IPv4

nodes = set()
while len(nodes) == 0 or not all((type(ip_address(host)) is target_type) for host in
    nodes):
    nodes = set()

    # This refreshes the cluster topology and will discovery any node updates.
    # Under the hood it calls cluster slots
    cluster.nodes_manager.initialize()
    for node in cluster.get_nodes():
        nodes.add(node.host)
    self.logger.info(nodes)

    time.sleep(1)
```

Esempio con Lettuce:

```
RedisClusterClient clusterClient = RedisClusterClient.create(RedisURI.create("xxxx",
    6379));

Class targetProtocolType = Inet6Address.class; // Or Inet4Address.class if you're
    switching to IPv4

Set<String> nodes;

do {
    // Check for any changes in the cluster topology.
    // Under the hood this calls cluster slots
    clusterClient.refreshPartitions();
```

```
Set<String> nodes = new HashSet<>();

for (RedisClusterNode node : clusterClient.getPartitions().getPartitions()) {
    nodes.add(node.getUri().getHost());
}

Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
    try {
        return finalTargetProtocolType.isInstance(InetAddress.getByName(node));
    } catch (UnknownHostException ignored) {}
    return false;
}));
```

Cluster dual stack abilitati per TLS ElastiCache

Quando TLS è abilitato per ElastiCache i cluster, le funzioni di rilevamento dei cluster (`cluster slotscluster shards`, `ecluster nodes`) restituiscono nomi host anziché IP. I nomi host vengono quindi utilizzati al posto degli IP per connettersi al ElastiCache cluster ed eseguire un handshake TLS. Ciò significa che i client non sono interessati dal parametro Individuazione IP. Per i cluster abilitati per TLS, il parametro Individuazione IP non ha alcun effetto sul protocollo IP preferito. Invece, il protocollo IP utilizzato verrà determinato in base a quello preferito dal client durante la risoluzione dei nomi host DNS.

Client Java

Quando si esegue la connessione da un ambiente Java che supporta sia IPv4 che IPv6, per impostazione predefinita Java preferirà IPv4 rispetto a IPv6 per la compatibilità con le versioni precedenti. Tuttavia, la preferenza del protocollo IP è configurabile tramite gli argomenti JVM. Per preferire IPv4, la JVM accetta `-Djava.net.preferIPv4Stack=true` e per preferire IPv6 imposta `-Djava.net.preferIPv6Stack=true`. Se si imposta `-Djava.net.preferIPv4Stack=true`, significa che la JVM non effettuerà più connessioni IPv6. Inclusi quelli per altre applicazioni non Redis.

Preferenze a livello di host

In generale, se il client o il runtime client non forniscono opzioni di configurazione per l'impostazione di una preferenza del protocollo IP, quando si esegue la risoluzione DNS, il protocollo IP dipenderà dalla configurazione dell'host. Per impostazione predefinita, la maggior parte degli host preferisce IPv6 rispetto a IPv4, ma questa preferenza può essere configurata a livello di host. Ciò influirà su tutte le richieste DNS provenienti da quell'host, non solo su quelle ai cluster. ElastiCache

Host Linux

Per Linux, una preferenza protocollo IP può essere configurata modificando il file `gai.conf`. Il file `gai.conf` è disponibile in `/etc/gai.conf`. Se non è specificato alcun `gai.conf`, uno di esempio deve essere disponibile in `/usr/share/doc/glibc-common-x.xx/gai.conf` che può essere copiato in `/etc/gai.conf`; è quindi necessario rimuovere i commenti dalla configurazione predefinita. Per aggiornare la configurazione in modo da preferire l'IPv4 durante la connessione a un ElastiCache cluster, aggiorna la precedenza per l'intervallo CIDR che comprende gli IP del cluster in modo che sia superiore alla precedenza per le connessioni IPv6 predefinite. Per impostazione predefinita, alle connessioni IPv6 viene assegnata una priorità di 40. Ad esempio, presupponendo che il cluster si trovi in una sottorete con il CIDR `172.31.0.0/16`, la configurazione sottostante porterebbe i client a preferire le connessioni IPv4 a tale cluster.

```
label ::1/128      0
label ::/0        1
label 2002::/16   2
label ::/96       3
label ::ffff:0:0/96 4
label fec0::/10   5
label fc00::/7    6
label 2001:0::/32 7
label ::ffff:172.31.0.0/112 8
#
# This default differs from the tables given in RFC 3484 by handling
# (now obsolete) site-local IPv6 addresses and Unique Local Addresses.
# The reason for this difference is that these addresses are never
# NATed while IPv4 site-local addresses most probably are. Given
# the precedence of IPv6 over IPv4 (see below) on machines having only
# site-local IPv4 and IPv6 addresses a lookup for a global address would
# see the IPv6 be preferred. The result is a long delay because the
# site-local IPv6 addresses cannot be used while the IPv4 address is
# (at least for the foreseeable future) NATed. We also treat Teredo
# tunnels special.
#
# precedence <mask> <value>
# Add another rule to the RFC 3484 precedence table. See section 2.1
# and 10.3 in RFC 3484. The default is:
#
precedence ::1/128      50
precedence ::/0        40
precedence 2002::/16   30
precedence ::/96       20
```

```
precedence ::ffff:0:0/96 10
precedence ::ffff:172.31.0.0/112 100
```

Ulteriori dettagli su `gai.conf` sono disponibili nella [pagina principale di Linux](#)

Host Windows

Il processo per gli host Windows è simile. Per gli host Windows è possibile eseguire `netsh interface ipv6 set prefix CIDR_CONTAINING_CLUSTER_IPS PRECEDENCE LABEL`. L'effetto è identico alla modifica del file `gai.conf` su host Linux.

Ciò aggiornerà le policy di preferenza in modo da preferire le connessioni IPv4 rispetto alle connessioni IPv6 per l'intervallo CIDR specificato. Ad esempio, presupponendo che il cluster si trovi in una sottorete con il CIDR `172.31.0.0/16`, l'esecuzione di `netsh interface ipv6 set prefix ::ffff:172.31.0.0:0/112 100 15` genererebbe la seguente tabella di priorità che porterebbe i client a preferire IPv4 durante la connessione al cluster.

```
C:\Users\Administrator>netsh interface ipv6 show prefixpolicies
Querying active state...
```

```
Precedence Label Prefix
-----
100 15 ::ffff:172.31.0.0:0/112
20 4 ::ffff:0:0/96
50 0 ::1/128
40 1 ::/0
30 2 2002::/16
5 5 2001::/32
3 13 fc00::/7
1 11 fec0::/10
1 12 3ffe::/16
1 3 ::/96
```

Best practice per l'utilizzo di cluster progettati autonomamente

Questa sezione si applica solo quando scegli di progettare i tuoi cluster Redis. Ti consigliamo di esaminare e seguire queste best practices.

Argomenti

- [Ridurre al minimo i tempi di inattività con la funzione Multi-AZ](#)
- [Verifica della disponibilità di memoria sufficiente per creare uno snapshot Redis](#)
- [Gestione della memoria prenotata](#)
- [Ridimensionamento di cluster online](#)
- [Riduzione al minimo dei tempi di inattività durante la manutenzione](#)

Ridurre al minimo i tempi di inattività con la funzione Multi-AZ

Per ulteriori informazioni su Multi-AZ e la riduzione dei tempi di inattività, consulta [Ridurre al minimo i tempi di inattività in ElastiCache per Redis con la funzione Multi-AZ](#).

Verifica della disponibilità di memoria sufficiente per creare uno snapshot Redis

Snapshot e sincronizzazioni Redis nella versione 2.8.22 e successive

Redis 2.8.22 introduce un processo di salvataggio senza fork che permette di allocare più memoria all'utilizzo dell'applicazione senza incorrere in un maggiore utilizzo di swap durante le sincronizzazioni e i salvataggi. Per ulteriori informazioni, consulta [Modalità di implementazione di sincronizzazione e backup](#).

Snapshot e sincronizzazioni Redis prima della versione 2.8.22

Quando utilizzi Redis ElastiCache, ecco i casi in cui Redis chiama un comando di scrittura in background:

- Durante la creazione di uno snapshot per un backup.
- Durante la sincronizzazione delle repliche con il nodo principale in un gruppo di repliche
- Durante l'abilitazione della caratteristica append-only file (AOF) per Redis.
- Durante la promozione di una replica a primaria (che causa una sincronizzazione tra nodi principali/replica).

Quando Redis esegue un processo di scrittura in background, devi disporre di memoria sufficiente per gestire il carico di elaborazione. La mancanza di memoria sufficiente provoca un errore del processo. Per questo motivo, è importante scegliere un tipo di istanza del nodo che abbia memoria sufficiente quando si crea il cluster Redis.

Processo di scrittura in background e utilizzo di memoria

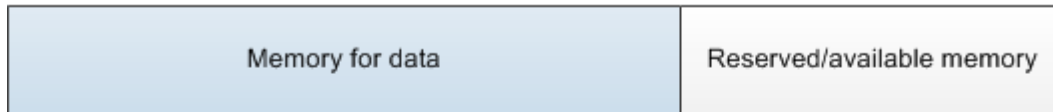
Ogni volta che viene chiamato un processo di scrittura in background, Redis esegue il fork del processo (Redis è a thread singolo). Un fork rende i dati persistenti sul disco, in un file snapshot RDB di Redis. Tutti gli altri servizi fork eseguono operazioni di lettura e scrittura. Per garantire che lo snapshot sia di tipo point-in-time, tutte le aggiunte e gli aggiornamenti dei dati vengono scritti in un'area di memoria disponibile separata dall'area dati.

Finché è disponibile memoria sufficiente per registrare tutte le operazioni di scrittura mentre i dati vengono resi persistenti sul disco, non si dovrebbero avere problemi di memoria insufficiente. Potresti riscontrare problemi di memoria insufficiente qualora si verifichi uno dei seguenti casi:

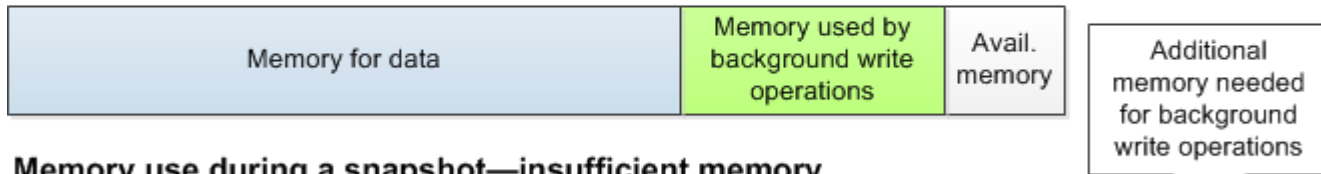
- L'applicazione esegue molte operazioni di scrittura e richiede pertanto un'elevata quantità di memoria disponibile per accettare dati nuovi o aggiornati.
- Hai a disposizione una quantità di memoria veramente limitata in cui scrivere dati nuovi o aggiornati.
- Il set di dati di cui disponi è di grandi dimensioni. Poiché per essere reso persistente sul disco è necessario molto tempo, sono necessarie numerose operazioni di scrittura.

Il diagramma seguente illustra l'utilizzo della memoria durante l'esecuzione di un processo di scrittura in background.

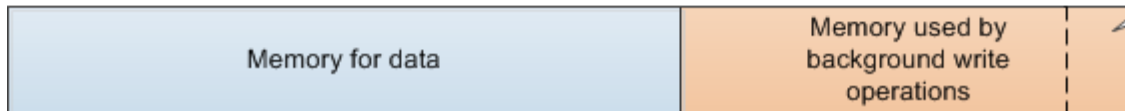
Memory use prior to a snapshot



Memory use during a snapshot—sufficient memory



Memory use during a snapshot—insufficient memory



Per informazioni sull'impatto di un backup sulle prestazioni, consulta [Impatto sulle prestazioni dei backup di cluster progettati autonomamente](#).

Per ulteriori informazioni sulla modalità di esecuzione degli snapshot da parte di Redis, consulta <http://redis.io>.

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

Come evitare di esaurire la memoria durante l'esecuzione di una scrittura in background

Quando viene chiamato un processo di scrittura in background, ad esempio BGSAVE o BGREWRITEAOF, per evitare che il processo abbia esito negativo, è necessario disporre di una quantità di memoria maggiore rispetto a quella che verrà utilizzata dalle operazioni di scrittura durante il processo. Il caso peggiore è rappresentato dallo scenario in cui durante l'operazione di scrittura in background ogni record Redis viene aggiornato e alcuni nuovi record vengono aggiunti alla cache. È pertanto consigliabile impostare `reserved-memory-percent` su 50 (50 per cento) per le versioni di Redis precedenti alla 2.8.22 o 25 (25 per cento) per le versioni di Redis 2.8.22 e successive.

Il valore `maxmemory` indica la memoria che hai a disposizione per i dati e il sovraccarico operativo. Poiché non puoi modificare il parametro `reserved-memory` nel gruppo di parametri di default, devi creare un gruppo di parametri personalizzato per il cluster. Il valore di default per `reserved-memory` è 0, che consente a Redis di utilizzare l'intero valore di `maxmemory` con i dati, lasciando una quantità di memoria insufficiente per altri utilizzi, ad esempio un processo di scrittura in background. Per i valori di `maxmemory` per tipo di istanza del nodo, consulta [Parametri specifici del tipo di nodo di Redis](#).

Puoi anche utilizzare il parametro `reserved-memory` per ridurre la quantità di memoria predefinita utilizzata da Redis.

Per ulteriori informazioni sui parametri specifici di Redis in ElastiCache, consulta [Parametri specifici di Redis](#).

Per informazioni sulla creazione e sulla modifica di gruppi di parametri, consulta [Creazione di un gruppo di parametri](#) e [Modifica di un gruppo di parametri](#).

Gestione della memoria prenotata

La memoria prenotata è la memoria destinata all'utilizzo diverso dai dati. Durante l'esecuzione di un backup o failover, Redis utilizza la memoria disponibile per registrare operazioni di scrittura nel cluster mentre i dati del cluster vengono scritti nel file `.rdb`. Se non disponi di memoria sufficiente per tutte le scritture, il processo non va a buon fine. Di seguito sono disponibili le informazioni sulle opzioni di gestione della memoria prenotata per ElastiCache for Redis e su come applicare tali opzioni.

Argomenti

- [Qual è la quantità di memoria prenotata necessaria?](#)
- [Parametri per gestire memoria prenotata](#)
- [Specifica del parametro di gestione della memoria prenotata](#)

Qual è la quantità di memoria prenotata necessaria?

Se stai eseguendo una versione di Redis precedente alla 2.8.22, prenota più memoria per i backup e i failover rispetto all'esecuzione di Redis 2.8.22 o versioni successive. Questo requisito è dovuto ai diversi modi utilizzati da ElastiCache for Redis per implementare il processo di backup. La regola generale è prenotare metà del valore `maxmemory` di un tipo di nodo per il sovraccarico Redis per le versioni precedenti alla 2.8.22 e un quarto per Redis versione 2.8.22 e successive.

Quando si utilizzano cluster con tiering di dati, si consiglia di aumentare la memoria massima fino a metà della memoria disponibile del nodo se il carico di lavoro è pesante in scrittura.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Verifica della disponibilità di memoria sufficiente per creare uno snapshot Redis](#)
- [Modalità di implementazione di sincronizzazione e backup](#)
- [Tiering di dati](#)

Parametri per gestire memoria prenotata

Dal 16 marzo 2017, Amazon ElastiCache for Redis fornisce due parametri mutuamente esclusivi per gestire la memoria Redis, `reserved-memory` e `reserved-memory-percent`. Nessuno di questi parametri fanno parte della distribuzione Redis.

A seconda di quando sei diventato un cliente ElastiCache, uno o l'altro di questi parametri è il parametro di gestione della memoria di default. Questo parametro si applica quando crei un nuovo cluster o gruppo di replica Redis e utilizzi un gruppo di parametri di default.

- Per i clienti che hanno iniziato prima del 16 marzo 2017 - Quando crei un cluster o gruppo di replica Redis utilizzando il gruppo di parametri di default, il parametro di gestione della memoria è `reserved-memory`. In questo caso, zero (0) byte di memoria sono riservati.
- Per i clienti che hanno iniziato il 16 marzo 2017 o in seguito - Quando crei un cluster o gruppo di replica Redis utilizzando il gruppo di parametri di default, il parametro di gestione della memoria è `reserved-memory-percent`. In questo caso, il 25% del valore `maxmemory` del nodo è riservato per scopi non dati.

Dopo aver letto i due parametri di gestione della memoria Redis, potresti preferire usare quello che non è il tuo di default o con valori non predefiniti. In questo caso, puoi passare all'altro parametro di gestione della memoria prenotata.

Per modificare il valore di tale parametro, è possibile creare un gruppo di parametri personalizzato e modificarlo in modo da utilizzare il parametro e il valore di gestione della memoria preferiti. Puoi quindi utilizzare il gruppo di parametri personalizzato ogni volta che crei un nuovo cluster o gruppo di replica Redis. Per cluster o gruppi di replica esistenti, puoi modificarli per utilizzare il gruppo di parametri personalizzato.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Specifica del parametro di gestione della memoria prenotata](#)
- [Creazione di un gruppo di parametri](#)
- [Modifica di un gruppo di parametri](#)
- [Modifica di un cluster ElastiCache](#)
- [Modifica di un gruppo di replica](#)

Il parametro `reserved-memory`

Prima del 16 marzo 2017, tutta la gestione della memoria prenotata di ElastiCache for Redis era eseguita utilizzando il parametro `reserved-memory`. Il valore di default di `reserved-memory` è 0. Questa impostazione predefinita non riserva memoria per l'overhead Redis e consente a Redis di utilizzare tutta la memoria di un nodo con i dati.

La modifica di `reserved-memory` per disporre di memoria sufficiente per i backup e i failover, richiede la creazione di un gruppo di parametri personalizzato. In questo gruppo di parametri personalizzato, imposta `reserved-memory` su un valore appropriato per la versione Redis in esecuzione sul cluster e il tipo di nodo del cluster. Per ulteriori informazioni, consulta [Qual è la quantità di memoria prenotata necessaria?](#)

Il parametro `reserved-memory` di ElastiCache for Redis è specifico per ElastiCache for Redis e non fa parte della distribuzione Redis.

Nella procedura seguente viene mostrato come utilizzare `reserved-memory` per gestire la memoria sul cluster Redis.

Per prenotare la memoria utilizzando la memoria prenotata

1. Creare un gruppo di parametri personalizzato specificando la famiglia di gruppi di parametri corrispondente alla versione del motore in esecuzione, ad esempio specificando la proprietà `redis2.8` famiglia di gruppi di parametri. Per ulteriori informazioni, consulta [Creazione di un gruppo di parametri](#).

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name redis6x-m3x1 \  
  --description "Redis 2.8.x for m3.xlarge node type" \  
  --cache-parameter-group-family redis6.x
```

2. Calcolare la quantità di byte di memoria che occorre prenotare per l'overhead Redis. Il valore di `maxmemory` per il tipo di nodo è disponibile in [Parametri specifici del tipo di nodo di Redis](#).
3. Modificare il gruppo di parametri personalizzato in modo che il parametro `reserved-memory` corrisponda al numero di byte calcolati nella fase precedente. L'esempio di AWS CLI seguente presuppone che sia in esecuzione una versione di Redis precedente alla 2.8.22 e la metà della `maxmemory` del nodo deve essere prenotata. Per ulteriori informazioni, consulta [Modifica di un gruppo di parametri](#).

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name redis28-m3x1 \  
  --parameter-name-values "ParameterName=reserved-memory,  
  ParameterValue=7130316800"
```

Occorre un gruppo di parametri personalizzato separato per ogni tipo di nodo in uso, perché ogni tipo di nodo dispone di un valore `maxmemory` diverso. Pertanto, ogni tipo di nodo richiede un valore diverso per `reserved-memory`.

4. Modificare il cluster o il gruppo di replica Redis per utilizzare il gruppo di parametri personalizzato.

L'esempio CLI seguente modifica il cluster `my-redis-cluster` per utilizzare da subito il gruppo di parametri personalizzato `redis28-m3x1`. Per ulteriori informazioni, consulta [Modifica di un cluster ElastiCache](#).

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-redis-cluster \  
  --cache-parameter-group-name redis28-m3x1 \  
  --apply-immediately
```

L'esempio CLI seguente modifica il gruppo di replica `my-redis-repl-grp` per utilizzare da subito il gruppo di parametri personalizzato `redis28-m3x1`. Per ulteriori informazioni, consult, [Modifica di un gruppo di replica](#).

```
aws elasticache modify-replication-group \  
  --replication-group-id my-redis-repl-grp \  
  --cache-parameter-group-name redis28-m3x1 \  
  --apply-immediately
```

Il parametro `reserved-memory-percent`

Il 16 marzo 2017, Amazon ElastiCache ha introdotto il parametro `reserved-memory-percent` e lo ha reso disponibile su tutte le versioni di ElastiCache for Redis. Lo scopo di `reserved-memory-percent` è semplificare la gestione della memoria prenotata tra tutti i cluster. A tal fine, l'utente può disporre di un singolo gruppo di parametri per ogni famiglia di gruppo di parametri (ad esempio `redis2.8`) per gestire la memoria prenotata dei cluster, a prescindere dal tipo di nodo. Il valore di default per `reserved-memory-percent` è 25 (25 per cento).

Il parametro `reserved-memory-percent` di ElastiCache for Redis è specifico di ElastiCache for Redis e non fa parte della distribuzione Redis.

Se il cluster utilizza un tipo di nodo della famiglia `r6gd` e l'utilizzo della memoria raggiunge il 75%, il tiering di dati verrà attivato automaticamente. Per ulteriori informazioni, consulta [Tiering di dati](#).

Per prenotare memoria utilizzando la percentuale di reserved-percent

Per utilizzare `reserved-memory-percent` per gestire la memoria sul cluster ElastiCache for Redis, esegui una delle seguenti operazioni:

- Se stai eseguendo Redis 2.8.22 o versioni successive, assegna semplicemente il gruppo di parametri di default al cluster. Il valore di default di 25 percento dovrebbe essere adeguato. In caso contrario, attenersi alla procedura descritta di seguito per modificare il valore.
- Se stai eseguendo una versione di Redis precedente alla 2.8.22, è probabile che sia necessario prenotare più memoria rispetto al 25% di default di `reserved-memory-percent`. A tale scopo, procedi come indicato di seguito.

Per modificare il valore percentuale di reserved-percent

1. Creare un gruppo di parametri personalizzato specificando la famiglia di gruppi di parametri corrispondente alla versione del motore in esecuzione, ad esempio specificando la famiglia di gruppi di parametri `redis2.8`. Un gruppo di parametri di default è necessario perché non è possibile modificare un gruppo di parametri di default. Per ulteriori informazioni, consulta [Creazione di un gruppo di parametri](#).

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name redis28-50 \  
  --description "Redis 2.8.x 50% reserved" \  
  --cache-parameter-group-family redis2.8
```

Poiché `reserved-memory-percent` consente di riservare memoria con una percentuale della `maxmemory` del nodo, non è necessario un gruppo di parametri personalizzato per ogni tipo di nodo.

2. Modificare il gruppo di parametri personalizzato in modo che `reserved-memory-percent` sia 50 (50 percento). Per ulteriori informazioni, consulta [Modifica di un gruppo di parametri](#).

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name redis28-50 \  
  --parameter-name-values "ParameterName=reserved-memory-percent,  
  ParameterValue=50"
```

3. Utilizzare questo gruppo di parametri personalizzato per qualsiasi cluster o gruppi di replica Redis che eseguono una versione di Redis precedente alla 2.8.22.

L'esempio CLI seguente modifica il cluster Redis `my-redis-cluster` per utilizzare da subito il gruppo di parametri personalizzato `redis28-50`. Per ulteriori informazioni, consulta [Modifica di un cluster ElastiCache](#).

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-redis-cluster \  
  --cache-parameter-group-name redis28-50 \  
  --apply-immediately
```

L'esempio CLI seguente modifica il gruppo di replica Redis `my-redis-repl-grp` per utilizzare da subito il gruppo di parametri personalizzato `redis28-50`. Per ulteriori informazioni, consulta [Modifica di un gruppo di replica](#).

```
aws elasticache modify-replication-group \  
  --replication-group-id my-redis-repl-grp \  
  --cache-parameter-group-name redis28-50 \  
  --apply-immediately
```

Specifica del parametro di gestione della memoria prenotata

Se il 16 marzo 2017 eri un cliente ElastiCache corrente, il parametro di gestione della memoria prenotata di default è `reserved-memory` con (0) byte di memoria prenotata. Se sei diventato un cliente ElastiCache dopo il 16 marzo 2017, il parametro di gestione della memoria prenotata di default è `reserved-memory-percent` con 25% di memoria prenotata del nodo. Questo è VERO a prescindere dalla data di creazione del cluster ElastiCache for Redis o del gruppo di replica. Tuttavia, puoi modificare il parametro di gestione della memoria prenotata, mediante la AWS CLI o l'API ElastiCache.

I parametri `reserved-memory` e `reserved-memory-percent` sono mutuamente esclusivi. Un gruppo di parametri ne conterrà sempre uno ma non entrambi. Puoi cambiare il parametro utilizzato da un gruppo di parametri per la gestione della memoria prenotata modificando il gruppo di parametri. Il gruppo di parametri deve essere un gruppo di parametri personalizzato, perché non puoi modificare i gruppi di parametri di default. Per ulteriori informazioni, consulta [Creazione di un gruppo di parametri](#).

Per specificare la percentuale `reserved-memory-percent`

Per utilizzare `reserved-memory-percent` come parametro di gestione della memoria prenotata, modifica un gruppo di parametri personalizzato utilizzando il comando `modify-cache-parameter-group`. Utilizzo del parametro `parameter-name-values` per specificare `reserved-memory-percent` e un valore per esso.

L'esempio CLI seguente modifica il gruppo di parametri personalizzato `redis32-cluster-on` in modo che utilizzi `reserved-memory-percent` per gestire memoria prenotata. Affinché il gruppo di parametri utilizzi il parametro `ParameterName` per la gestione della memoria prenotata, occorre assegnare un valore a `ParameterValue`. Per ulteriori informazioni, consulta [Modifica di un gruppo di parametri](#).

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name redis32-cluster-on \  
  --parameter-name-values "ParameterName=reserved-memory-percent, ParameterValue=25"
```

Per specificare la memoria `reserved-memory`

Per utilizzare `reserved-memory` come parametro di gestione della memoria prenotata, modifica un gruppo di parametri personalizzato utilizzando il comando `modify-cache-parameter-group`. Utilizzo del parametro `parameter-name-values` per specificare `reserved-memory` e un valore per esso.

L'esempio CLI seguente modifica il gruppo di parametri personalizzato `redis32-m3x1` in modo che utilizzi `reserved-memory` per gestire memoria prenotata. Affinché il gruppo di parametri utilizzi il parametro `ParameterName` per la gestione della memoria prenotata, occorre assegnare un valore a `ParameterValue`. Poiché la versione del motore è più recente di 2.8.22, impostiamo il valore su `3565158400` che è 25 per cento di una `maxmemory` di cache `.m3.xlarge`. Per ulteriori informazioni, consulta [Modifica di un gruppo di parametri](#).

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name redis32-m3x1 \  
  --parameter-name-values "ParameterName=reserved-memory, ParameterValue=3565158400"
```

Ridimensionamento di cluster online

Il `resharding` implica l'aggiunta e la rimozione di partizioni o nodi nel cluster e la ridistribuzione 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 attenerti 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. È dunque consigliabile mantenere l'utilizzo della CPU sotto l'80% sulle istanze multicore e sotto il 50% sulle istanze a core singolo durante il resharding. Esegui il monitoraggio dei parametri di ElastiCache 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 `BytesUsedForCacheItems`.
- 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.
- Carica le tue funzioni su ogni partizione - Quando si aumenta orizzontalmente il cluster, ElastiCache replicherà automaticamente le funzioni caricate in uno dei nodi esistenti (selezionati casualmente) nei nuovi nodi. Se il cluster dispone di Redis 7.0 o versioni successive e l'applicazione utilizza [Funzioni Redis](#), consigliamo di caricare tutte le funzioni in tutte le partizioni prima di aumentare orizzontalmente il cluster in modo che non contenga funzioni diverse su partizioni differenti.

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. I dati sulle partizioni di destinazione non verranno eliminati.
- 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.
- I comandi FLUSHALL e FLUSHDB non sono supportati negli script Lua durante un'operazione di riassegnazione delle partizioni. Nelle versioni precedenti a Redis 6, il comando BRPOPLPUSH non è supportato se eseguito sullo slot da migrare.

Riduzione al minimo dei tempi di inattività durante la manutenzione

La configurazione in modalità cluster ha la migliore disponibilità durante le operazioni gestite o non gestite. Si consiglia di utilizzare un client supportato in modalità cluster che si connette all'endpoint di individuazione del cluster. Per la modalità cluster disattivata, si consiglia di utilizzare l'endpoint primario per tutte le operazioni di scrittura.

Per attività di lettura, le applicazioni possono inoltre connettersi a qualsiasi nodo nel cluster. A differenza dell'endpoint primario, gli endpoint dei nodi vengono risolti in endpoint specifici. Se apporti una modifica al cluster, ad esempio aggiungendo o eliminando una replica, devi aggiornare gli endpoint dei nodi nell'applicazione. Questo è il motivo per cui, con la modalità cluster disabilitata, ti consigliamo di utilizzare l'endpoint di lettura per le attività di lettura.

Se il failover automatico è abilitato nel cluster, il nodo primario potrebbe cambiare. Pertanto, l'applicazione deve confermare il ruolo del nodo e aggiornare tutti gli endpoint di lettura. In questo modo hai la garanzia che non verrà causato un carico eccessivo sull'endpoint primario. Con il failover automatico disabilitato, il ruolo del nodo non cambia. Tuttavia, il tempo di inattività nelle operazioni gestite o non gestite è superiore rispetto ai cluster con failover automatico abilitato.

Evita di indirizzare le richieste di lettura a un singolo nodo di replica di lettura perché la sua indisponibilità potrebbe causare un'interruzione della lettura. Esegui il fallback alla lettura del nodo primario oppure assicurati di disporre di almeno due repliche di lettura per evitare interruzioni di lettura durante la manutenzione.

Best practice per Redis

Di seguito sono riportate le best practice per l'utilizzo di Redis per migliorare le prestazioni e l'affidabilità:

- Usa configurazioni con la modalità cluster abilitata: la modalità cluster abilitata consente alla cache di dimensionarsi orizzontalmente per ottenere un'archiviazione e una velocità di trasmissione effettiva più elevati rispetto a una configurazione con la modalità cluster disabilitata. ElastiCache serverless è disponibile solo in una configurazione con la modalità cluster abilitata.
- Usa connessioni di lunga durata: la creazione di una nuova connessione è costosa e richiede tempo e risorse della CPU della cache. Riutilizza le connessioni quando possibile (ad esempio con il pool di connessioni) per ammortizzare questo costo rispetto a molti comandi.
- Leggi dalle repliche: se utilizzi ElastiCache serverless o hai eseguito il provisioning delle repliche di lettura (cluster progettati autonomamente), indirizza le letture alle repliche per ottenere una migliore scalabilità e/o una latenza inferiore. Le letture dalle repliche sono consistenti finali con il nodo primario.

In un cluster progettato autonomamente, evita di indirizzare le richieste di lettura a una singola replica di lettura perché le letture potrebbero non essere temporaneamente disponibili in caso di errore del nodo. Puoi configurare il client per indirizzare le richieste di lettura ad almeno due repliche di lettura oppure indirizzare le letture a una sola replica e al nodo primario.

In ElastiCache serverless, la lettura dalla porta di replica (6380) indirizza le letture alla zona di disponibilità locale del client, quando possibile, riducendo la latenza di recupero. Esegue automaticamente il fallback agli altri nodi in caso di errore.

- 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. Sugeriamo 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.
- Usa pub/sub partizionato: [quando utilizzi Redis per supportare carichi di lavoro pub/sub con velocità di trasmissione effettiva elevata, ti consigliamo di utilizzare pub/sub partizionato](#) (disponibile con Redis 7 o versioni successive). I cluster pub/sub tradizionali con la modalità cluster abilitata trasmettono messaggi a tutti i nodi del cluster, il che può comportare un aumento

di `EngineCPUUtilization`. Tieni presente che in ElastiCache serverless, i comandi pub/sub tradizionali utilizzano internamente comandi pub/sub partizionati.

Strategie di caching

Nel seguente argomento, puoi trovare strategie per popolare e mantenere la cache.

L'implementazione di strategie per il popolamento e la gestione della cache dipende dal tipo di dati che desideri memorizzare e dai modelli di accesso a tali dati. Ad esempio, probabilmente non utilizzerai la stessa strategia per la classifica dei 10 migliori punteggi di gioco in un sito di gaming, o per le notizie più interessanti. Nella parte restante di questa sezione, discuteremo delle strategie più comuni di gestione della cache, dei loro vantaggi e svantaggi.

Argomenti

- [Caricamento lento](#)
- [Write-Through](#)
- [Aggiunta di TTL](#)
- [Argomenti correlati](#)

Caricamento lento

Come dice lo stesso nome, il caricamento lento è una strategia di caching che consente di caricare i dati nella cache solo quando necessario. Funziona come descritto di seguito.

Amazon ElastiCache è un archivio chiave-valore in memoria che si colloca tra la tua applicazione e il datastore (database) ai quali può accedere. Ogni qualvolta la tua applicazione richiama dati, esso invia per prima cosa la richiesta alla cache ElastiCache. Se i dati sono presenti nella cache e sono correnti, ElastiCache restituisce tali dati all'applicazione. Se i dati non sono presenti nella cache o sono scaduti, l'applicazione richiede i dati al Data Store. L'archivio dati restituisce quindi i dati all'applicazione. Successivamente, l'applicazione scrive i dati ricevuti dall'archivio nella cache. In questo modo, può essere recuperato più rapidamente la prossima volta che viene richiesto.

Un'occorrenza della cache si verifica quando i dati sono presenti nella cache e non sono scaduti:

1. L'applicazione richiede i dati alla cache.
2. La cache restituisce i dati all'applicazione.

Una mancato riscontro nella cache si verifica quando i dati non sono presenti nella cache o sono scaduti:

1. L'applicazione richiede i dati alla cache.
2. La cache non dispone dei dati richiesti e restituisce un `null`.
3. L'applicazione richiede e riceve i dati dal database.
4. L'applicazione aggiorna la cache con i nuovi dati.

Vantaggi e svantaggi del caricamento lento

I vantaggi del caricamento lento sono i seguenti:

- Solo i dati richiesti vengono memorizzati nella cache.

Poiché la maggior parte dei dati non viene mai richiesta, il caricamento lento evita di riempire la cache con i dati non richiesti.

- Gli errori dei nodi non sono fatali per l'applicazione.

Se un nodo restituisce un errore e viene sostituito da un nuovo nodo vuoto, l'applicazione continua a funzionare, nonostante l'aumento della latenza. Quando le richieste vengono effettuate al nuovo nodo, ogni mancato riscontro nella cache comporta una query del database. Allo stesso tempo, la copia dei dati viene aggiunta alla cache in modo che le richieste successive vengano richiamate dalla cache.

Gli svantaggi del caricamento pigro sono i seguenti:

- Comporta una penalità per mancato riscontro nella cache. Ogni mancato riscontro nella cache comporta tre passaggi:
 1. Richiesta iniziale dei dati dalla cache
 2. Query del database per i database
 3. Scrittura dei dati sulla cache

Questi mancati riscontri possono causare un notevole ritardo dei dati destinati all'applicazione.

- Dati obsoleti.

Se i dati vengono scritti nella cache solo quando si verifica un mancato riscontro nella cache, i dati nella cache diventano obsoleti. Questo risultato si verifica perché non sono stati aggiornati alla cache quando i dati vengono modificati nel database. Per risolvere questo problema, è possibile utilizzare il [Write-Through](#) e strategie [Aggiunta di TTL](#).

Esempio di pseudocodice di caricamento pigro

Il codice seguente è un esempio di pseudo codice della logica di caricamento lento.

```
// *****
// function that returns a customer's record.
// Attempts to retrieve the record from the cache.
// If it is retrieved, the record is returned to the application.
// If the record is not retrieved from the cache, it is
//   retrieved from the database,
//   added to the cache, and
//   returned to the application
// *****
get_customer(customer_id)

    customer_record = cache.get(customer_id)
    if (customer_record == null)

        customer_record = db.query("SELECT * FROM Customers WHERE id = {0}",
customer_id)
        cache.set(customer_id, customer_record)

    return customer_record
```

Per questo esempio, il codice dell'applicazione che ottiene i dati è il seguente.

```
customer_record = get_customer(12345)
```

Write-Through

La strategia di scrittura contemporanea aggiunge i dati o li aggiorna nella cache ogni qualvolta i dati vengano scritti sul database.

Vantaggi e svantaggi della scrittura contemporanea

I vantaggi di write-through sono i seguenti:

- I dati nella cache non sono mai obsoleti.

Poiché i dati nella cache vengono aggiornati ogni volta che vengono scritti sul database, i dati nella cache sono sempre correnti.

- Penalità di scrittura e penalità di lettura.

Ogni scrittura comporta due passaggi:

1. Una scrittura sulla cache
2. Una scrittura sul database

Che aggiunge latenza al processo. Detto questo, gli utenti finali sono generalmente più tolleranti della latenza durante l'aggiornamento o il richiamo dei dati. Esiste la convinzione che gli aggiornamenti comportino maggior lavoro e richiedano pertanto tempi lunghi.

Gli svantaggi di write-through sono i seguenti:

- Dati mancanti.

Se si esegue un giro verso l'alto di un nuovo nodo, dovuto a un errore del nodo o a un dimensionamento orizzontale, i dati mancanti sono stati visualizzati. Questi dati continuano a mancare fino a quando non vengono aggiunti o aggiornati nel database. È possibile ridurre al minimo questo implementando [caricamento lento](#) con write-through.

- Abbandono della cache.

La maggior parte dei dati non viene mai letta, il che è uno spreco di risorse. Da [aggiunta di un valore durata \(TTL\)](#), è possibile ridurre gli sprechi di spazio.

Esempio di pseudocodice write-through

Il codice seguente è un esempio di pseudo codice della logica di wright-through.

```
// *****  
// function that saves a customer's record.  
// *****  
save_customer(customer_id, values)  
  
    customer_record = db.query("UPDATE Customers WHERE id = {0}", customer_id, values)  
    cache.set(customer_id, customer_record)  
    return success
```

Per questo esempio, il codice dell'applicazione che ottiene i dati è il seguente.

```
save_customer(12345, {"address": "123 Main"})
```

Aggiunta di TTL

Il caricamento lento accetta i dati obsoleti, ma non restituisce errori con i nodi vuoti. La scrittura contemporanea garantisce dati sempre aggiornati ma può restituire un errore con i nodi vuoti e può popolare la cache con dati superflui. Aggiungendo un valore durata (TTL) a ogni scrittura, puoi avere i vantaggi di ogni strategia. Allo stesso tempo, puoi in gran parte evitare di ingombrare la cache con dati extra.

La durata (TTL) è un valore intero che specifica il numero di secondi fino alla scadenza della chiave. Redis può specificare secondi o millisecondi per questo valore. Quando un'applicazione tenta di leggere una chiave scaduta, viene trattata come se la chiave non fosse mai stata trovata. Il database viene interrogato per la chiave e la cache viene aggiornata. Questo approccio non garantisce che un valore non sia obsoleto. Tuttavia, impedisce ai dati di diventare troppo obsoleti e richiede che i valori nella cache vengano occasionalmente aggiornati dal database.

Per ulteriori informazioni, consulta il comando [Redis set](#) .

Esempi di pseudocodice TTL

Il codice seguente è un esempio di pseudo codice della logica di write-through con TTL.

```
// *****  
// function that saves a customer's record.  
// The TTL value of 300 means that the record expires  
// 300 seconds (5 minutes) after the set command  
// and future reads will have to query the database.  
// *****  
save_customer(customer_id, values)  
  
    customer_record = db.query("UPDATE Customers WHERE id = {0}", customer_id, values)  
    cache.set(customer_id, customer_record, 300)  
  
return success
```

Il codice seguente è un esempio di pseudo codice della logica di caricamento lento con TTL.

```
// *****  
// function that returns a customer's record.  
// Attempts to retrieve the record from the cache.
```

```
// If it is retrieved, the record is returned to the application.
// If the record is not retrieved from the cache, it is
//   retrieved from the database,
//   added to the cache, and
//   returned to the application.
// The TTL value of 300 means that the record expires
//   300 seconds (5 minutes) after the set command
//   and subsequent reads will have to query the database.
// *****
get_customer(customer_id)

    customer_record = cache.get(customer_id)

    if (customer_record != null)
        if (customer_record.TTL < 300)
            return customer_record          // return the record and exit function

    // do this only if the record did not exist in the cache OR
    //   the TTL was >= 300, i.e., the record in the cache had expired.
    customer_record = db.query("SELECT * FROM Customers WHERE id = {0}", customer_id)
    cache.set(customer_id, customer_record, 300) // update the cache
    return customer_record          // return the newly retrieved record and exit
function
```

Per questo esempio, il codice dell'applicazione che ottiene i dati è il seguente.

```
save_customer(12345, {"address": "123 Main"})
```


```
customer_record = get_customer(12345)
```

Argomenti correlati

- [Datastore in memoria](#)
- [Scelta di un motore e di una versione](#)
- [Scalabilità per Redis ElastiCache](#)

Gestione del cluster progettato autonomamente

Questa sezione include gli argomenti che ti aiutano a gestire i cluster progettati autonomamente.

 Note

Questi argomenti non si applicano a ElastiCache serverless.

Argomenti

- [Auto Scaling ElastiCache per cluster Redis](#)
- [Modifica della modalità cluster](#)
- [Replica tra regioni AWS mediante Global Datastore](#)
- [Alta disponibilità utilizzando gruppi di replica](#)
- [Gestione della manutenzione](#)
- [Configurazione dei parametri di motore con i gruppi di parametri](#)

Auto Scaling ElastiCache per cluster Redis

Prerequisiti

ElastiCache per Redis Auto Scaling è limitato a quanto segue:

- Cluster Redis (modalità cluster abilitata) che eseguono il motore Redis dalla versione 6.0 e successive
- Cluster di suddivisione in livelli dei dati (modalità cluster abilitata) che eseguono il motore Redis dalla versione 7.0.7 e successive
- Dimensioni istanza - Large, XL, 2xLarge
- Famiglie di tipi di istanza: R7g, R6g, R6gd, R5, M7g, M6g, M5, C7gn
- Auto Scaling in ElastiCache for Redis non è supportato per i cluster in esecuzione in Datastore globali, Outposts o Local Zones.

Gestione automatica della capacità con ElastiCache for Redis Auto Scaling

ElastiCache per Redis la scalabilità automatica è la capacità di aumentare o diminuire automaticamente gli shard o le repliche desiderati nel servizio ElastiCache for Redis. ElastiCache for Redis sfrutta il servizio Application Auto Scaling per fornire questa funzionalità. Per ulteriori informazioni, consulta [Application Auto Scaling](#). Per utilizzare il ridimensionamento automatico, è

necessario definire e applicare una politica di ridimensionamento che utilizza CloudWatch metriche e valori target assegnati dall'utente. ElastiCache for Redis auto scaling utilizza la policy per aumentare o diminuire il numero di istanze in risposta ai carichi di lavoro effettivi.

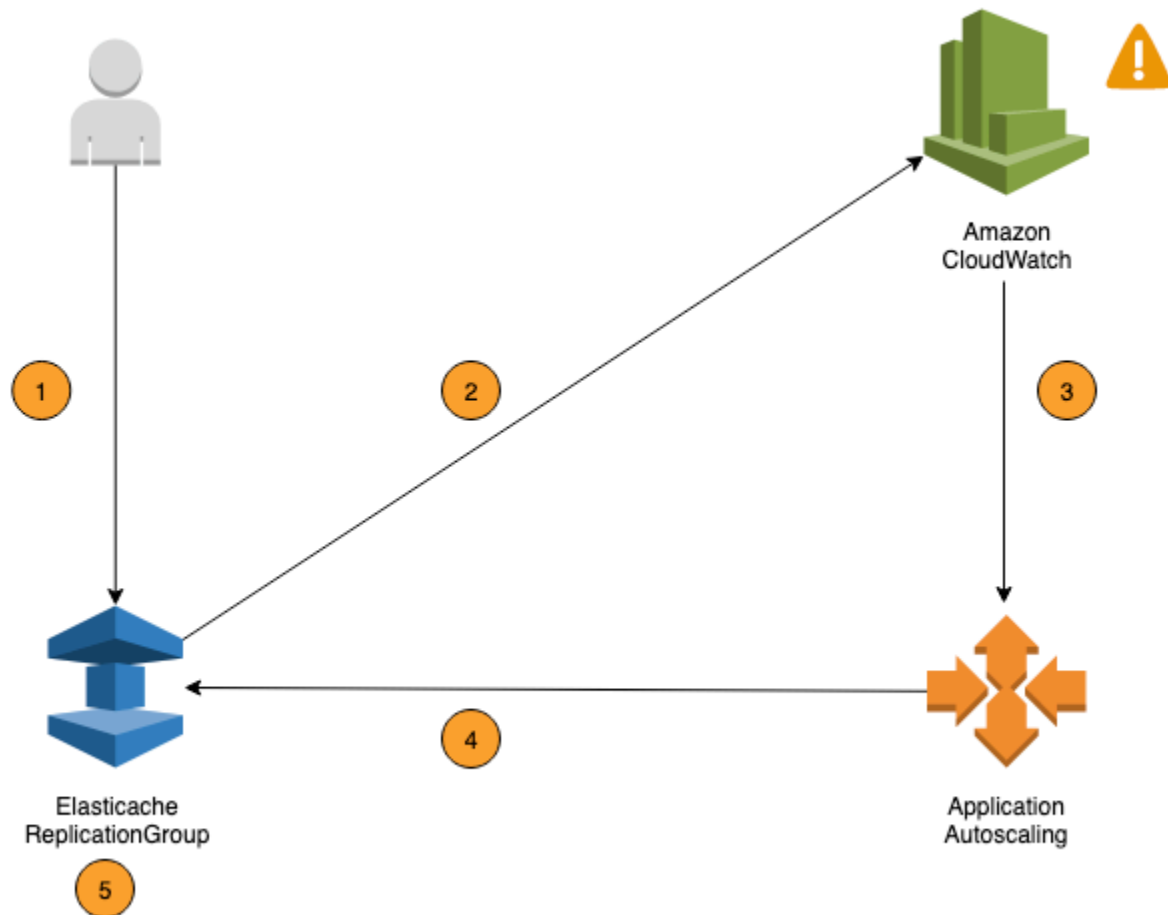
È possibile utilizzare il AWS Management Console per applicare una politica di scalabilità basata su una metrica predefinita. Un `predefined metric` viene definito in un'enumerazione in modo che tu possa specificarlo per nome nel codice o utilizzarlo nella AWS Management Console. I parametri personalizzati non possono essere selezionati tramite AWS Management Console. In alternativa, puoi utilizzare l'API Application Auto Scaling AWS CLI o l'API Application Auto Scaling per applicare una politica di scalabilità basata su una metrica predefinita o personalizzata.

ElastiCache for Redis supporta il ridimensionamento per le seguenti dimensioni:

- **Partizioni** : Aggiungi/rimuovi automaticamente frammenti nel cluster in modo simile al risharding online manuale. In questo caso, ElastiCache per Redis la scalabilità automatica attiva la scalabilità per tuo conto.
- **Repliche** : Aggiungi/rimuovi automaticamente le repliche nel cluster in modo simile alle operazioni manuali Aumenta/Diminuisci. ElastiCache for Redis auto scaling aggiunge/rimuove le repliche in modo uniforme su tutti gli shard del cluster.

ElastiCache for Redis supporta i seguenti tipi di politiche di scalabilità automatica:

- [Policy di dimensionamento con monitoraggio degli obiettivi](#): aumentare o diminuire il numero di attività eseguite dal servizio in base al valore di destinazione per un parametro specifico. Questa operazione può essere paragonata al modo in cui il termostato regola la temperatura di una casa. Tu selezioni la temperatura, il termostato si occupa del resto.
- [Scalabilità pianificata per Application ElastiCache for Redis auto Scaling](#): aumenta o diminuisci il numero di shard/repliche eseguite dal servizio in base alla data e all'ora.



I seguenti passaggi riassumono il processo di scalabilità automatica ElastiCache per Redis come mostrato nel diagramma precedente:

1. Crei una policy di auto scaling ElastiCache for Redis per il tuo ElastiCache for Redis Replication Group.
2. ElastiCache for Redis auto scaling crea un paio di CloudWatch allarmi per tuo conto. Ogni coppia rappresenta i tuoi limiti superiore e inferiore per i parametri. Questi CloudWatch allarmi vengono attivati quando l'utilizzo effettivo del cluster si discosta dall'utilizzo previsto per un periodo di tempo prolungato. Puoi ora visualizzare gli allarm nella console .
3. Se il valore della metrica configurata supera l'utilizzo previsto (o scende al di sotto del target) per un periodo di tempo specifico, CloudWatch attiva un allarme che richiama la scalabilità automatica di Redis ElastiCache per valutare la politica di scalabilità.
4. ElastiCache for Redis auto scaling emette una richiesta di modifica per regolare la capacità del cluster.

5. ElastiCache for Redis elabora la richiesta di modifica, aumentando (o diminuendo) dinamicamente la capacità degli shard/repliche del cluster in modo che si avvicini all'utilizzo previsto.

Per capire come ElastiCache funziona Redis Auto Scaling, supponiamo di avere un cluster denominato `UsersCluster`. Monitorando le CloudWatch metriche per `UsersCluster`, si determina il numero massimo di shard richiesti dal cluster quando il traffico è al suo picco e il numero minimo di shard quando il traffico è nel punto più basso. È inoltre possibile decidere un valore di destinazione per l'utilizzo della CPU per il cluster `UsersCluster`. ElastiCache for Redis auto scaling utilizza il suo algoritmo di tracciamento del target per garantire che gli shard di `UsersCluster` forniti vengano regolati secondo necessità in modo che l'utilizzo rimanga pari o vicino al valore target.

Note

La scalabilità potrebbe richiedere molto tempo e richiederà risorse di cluster aggiuntive per il ribilanciamento degli shard. ElastiCache for Redis Auto Scaling modifica le impostazioni delle risorse solo quando il carico di lavoro effettivo rimane elevato (o depresso) per un periodo prolungato di diversi minuti. L'algoritmo di tracciamento del target con scalabilità automatica di ElastiCache for Redis cerca di mantenere l'utilizzo del target pari o vicino al valore scelto a lungo termine.

Policy di Auto Scaling

Una policy di dimensionamento dispone dei seguenti componenti:

- Un parametro di destinazione : Il parametro CloudWatch che ElastiCache che per Redis Auto Scaling utilizza per determinare quando e quanto dimensionare.
- Capacità minima e massima: il numero minimo e massimo di shard o repliche da utilizzare per il dimensionamento.

Important

Durante la creazione della policy di Auto Scaling, se la capacità corrente è superiore alla capacità massima configurata, verrà ridotta orizzontalmente alla `MaxCapacity` durante la creazione della policy. Allo stesso modo, se la capacità corrente è inferiore alla capacità minima configurata, verrà aumentata orizzontalmente alla `MinCapacity`.

- Un tempo di raffreddamento: la quantità di tempo, in secondi, dopo il completamento di un'attività per ridurre orizzontalmente o aumentare orizzontalmente prima che un'altra attività per aumentare orizzontalmente si possa avviare.
- Un ruolo collegato al servizio — Un ruolo di Identity and Access Management (IAM) AWS collegato a uno specifico servizio AWS. Un ruolo collegato al servizio include tutte le autorizzazioni delle quali il servizio ha bisogno per chiamare altri servizi AWS per tuo conto. ElastiCache for Redis Auto Scaling genera in automatico questo ruolo, `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG`, per te.
- Attiva o disabilita le attività di riduzione orizzontale: capacità di abilitare o disabilitare le attività di riduzione orizzontale per una policy.

Argomenti

- [Parametro di destinazione per Auto Scaling](#)
- [Capacità minima e massima](#)
- [Periodo di raffreddamento](#)
- [Abilitazione o disabilitazione delle attività di riduzione](#)

Parametro di destinazione per Auto Scaling

In questo tipo di policy, vengono specificati un parametro personalizzato o predefinito e un valore di destinazione per il parametro in una configurazione della policy di dimensionamento con monitoraggio degli obiettivi. ElastiCache for Redis Auto Scaling crea e gestisce gli allarmi di CloudWatch per l'attivazione della policy di dimensionamento e calcola gli adeguamenti di dimensionamento in base al parametro e al valore di destinazione. La policy di dimensionamento aggiunge o rimuove le shard/repliche come richiesto per mantenere il parametro al valore di destinazione specificato o vicino a esso. Oltre a mantenere il parametro vicino al valore di destinazione, una policy di dimensionamento con monitoraggio degli obiettivi si adatta anche alle oscillazioni del parametro dovute a un carico di lavoro mutevole. Tale policy riduce anche le fluttuazioni rapide nel numero di shard/repliche disponibili per il cluster.

Ad esempio, prendi una policy di dimensionamento che usa il parametro medio di default `ElastiCachePrimaryEngineCPUUtilization`. Tale policy può mantenere l'utilizzo della CPU a una percentuale specifica di utilizzo, come il 70 per cento, o vicino ad essa.

 Note

Per ogni cluster , è possibile creare solo una policy di Auto Scaling per ogni parametro di destinazione.

Capacità minima e massima


Shard

Puoi specificare il numero massimo di shard che può essere scalato da ElastiCache per la scalabilità automatica Redis. Questo valore deve essere uguale o inferiore a 250 con un minimo di 1. Puoi anche specificare il numero minimo di shard che deve essere gestito dalla scalabilità automatica di ElastiCache for Redis. Questo valore deve essere almeno 1 e o uguale o minore del valore specificato per il numero massimo di 250 shard.

Repliche

È possibile specificare il numero massimo di repliche che devono essere gestite da ElastiCache for Redis Auto Scaling. Questo valore deve essere maggiore o uguale a 5. È anche possibile specificare il numero minimo di repliche che devono essere gestite dalla scalabilità automatica di ElastiCache for Redis. Questo valore deve essere almeno 1 e o uguale o inferiore del valore 5 specificato per il numero massimo repliche.

Per determinare il numero minimo e massimo di shard/repliche necessarie per il traffico tipico, esegui il test della configurazione di Auto Scaling con il tasso di traffico previsto per il modello.

 Note

I criteri di ridimensionamento automatico ElastiCache for Redis aumentano la capacità del cluster fino a raggiungere la dimensione massima definita o fino a quando non si applicano i limiti di servizio. Per richiedere un aumento dei limiti, consulta [AWSLimiti del servizio](#) e seleziona il tipo di limite Nodi per cluster per tipo di istanza.

 Important

Ridurre orizzontalmente in assenza di traffico Se il traffico di una variante diventa zero, ElastiCache for Redis si riduce automaticamente al numero minimo di istanze specificato.

Periodo di raffreddamento

È possibile sintonizzare i tempi di risposta di una policy di dimensionamento con monitoraggio degli obiettivi aggiungendo dei tempi di raffreddamento che influiscano sul dimensionamento del cluster. Un periodo di attesa blocca le richieste di riduzione o aumento ulteriori finché il periodo non scade. Questo rallenta le eliminazioni di shard o repliche nel cluster ElastiCache for Redis per le richieste di riduzione e la creazione di shard o repliche per le richieste di aumento. Puoi specificare i seguenti periodi di attesa:

- Un'attività di riduzione orizzontale riduce il numero di shard/repliche nel cluster ElastiCache for Redis. Un periodo di attesa di riduzione specifica la quantità di tempo che deve passare, in secondi, tra il completamento di un'attività di riduzione e l'inizio di un'altra attività di questo tipo.
- Un'attività di aumento orizzontale aumenta il numero di shard/repliche nel cluster ElastiCache for Redis. Un periodo di attesa di aumento specifica la quantità di tempo che deve passare, in secondi, tra il completamento di un'attività di aumento e l'inizio di un'altra attività di questo tipo.

Quando un periodo di raffreddamento di riduzione orizzontale o di aumento orizzontale non viene specificato, il valore di default per ogni aumento orizzontale è pari a 600 secondi e 900 per ogni riduzione orizzontale.

Abilitazione o disabilitazione delle attività di riduzione

Puoi abilitare o disabilitare le attività di riduzione per una policy. Abilitare queste attività di riduzione orizzontale consente alla policy di dimensionamento di eliminare le repliche /shard. Quando le attività di riduzione sono abilitate, il periodo di attesa della riduzione nella policy di dimensionamento si applica alle attività di riduzione. Disabilitare le attività di riduzione orizzontale evita alla policy di dimensionamento di eliminare le repliche /shard.

Note

Le attività di aumento orizzontale sono sempre abilitate, in modo che la policy di dimensionamento possa creare repliche ElastiCache for Redis in base alle esigenze.

Autorizzazioni IAM richieste ElastiCache per Redis Auto Scaling

ElastiCache for Redis Auto Scaling è reso possibile da una combinazione delle ElastiCache API for Redis CloudWatch e Application Auto Scaling. I cluster vengono creati e aggiornati con ElastiCache for Redis, gli allarmi vengono creati e le politiche di scalabilità vengono create con CloudWatch

Application Auto Scaling. Oltre alle autorizzazioni IAM standard per la creazione e l'aggiornamento dei cluster, l'utente IAM che accede alle ElastiCache impostazioni Redis Auto Scaling deve disporre delle autorizzazioni appropriate per i servizi che supportano la scalabilità dinamica. Gli utenti IAM devono disporre dell'autorizzazione per utilizzare le operazioni nella seguente policy di esempio:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:*",
        "elasticache:DescribeReplicationGroups",
        "elasticache:ModifyReplicationGroupShardConfiguration",
        "elasticache:IncreaseReplicaCount",
        "elasticache:DecreaseReplicaCount",
        "elasticache:DescribeCacheClusters",
        "elasticache:DescribeCacheParameters",
        "cloudwatch:DeleteAlarms",
        "cloudwatch:DescribeAlarmHistory",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:DescribeAlarmsForMetric",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:DisableAlarmActions",
        "cloudwatch:EnableAlarmActions",
        "iam:CreateServiceLinkedRole",
        "sns:CreateTopic",
        "sns:Subscribe",
        "sns:Get*",
        "sns:List*"
      ],
      "Resource": "arn:aws:iam::123456789012:role/autoscaling-roles-for-cluster"
    }
  ]
}
```

Ruolo collegato al servizio

Il servizio di auto scaling ElastiCache for Redis necessita inoltre dell'autorizzazione per descrivere i cluster e gli CloudWatch allarmi e delle autorizzazioni per modificare la capacità

di destinazione di ElastiCache for Redis per tuo conto. Se abiliti Auto Scaling per il tuo cluster ElastiCache for Redis, viene creato un ruolo collegato al servizio denominato `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG`. Questo ruolo legato al servizio concede a ElastiCache Redis auto scaling l'autorizzazione a descrivere gli allarmi per le tue politiche, monitorare la capacità attuale della flotta e modificare la capacità della flotta. Il ruolo collegato al servizio è il ruolo predefinito ElastiCache per la scalabilità automatica di Redis. Per ulteriori informazioni, consulta [Ruoli collegati ai servizi ElastiCache per la scalabilità automatica Redis nella Application Auto Scaling User Guide](#).

Best practice Auto Scaling

Prima di effettuare la registrazione a Auto Scaling, ti consigliamo di attenerci alle seguenti indicazioni:

1. Usa una sola metrica di tracciamento: consente di identificare se il cluster dispone di carichi di lavoro che utilizzano notevoli risorse di CPU o dati e di utilizzare la metrica predefinita corrispondente per definire la policy di dimensionamento.
 - CPU del motore: `ElastiCachePrimaryEngineCPUUtilization` (dimensione partizione) o `ElastiCacheReplicaEngineCPUUtilization` (dimensione replica)
 - Utilizzo del database:
`ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` questa policy di dimensionamento è ideale con `maxmemory-policy` impostato su `noeviction` sul cluster.

Ti consigliamo di evitare più policy per dimensione nel cluster. ElastiCache per Redis Auto scaling ridimensionerà la destinazione scalabile se alcune policy di tracciamento della destinazione sono pronte per la scalabilità orizzontale, ma la scalerà solo se tutte le politiche di tracciamento della destinazione (con la parte scalabile abilitata) sono pronte per la scalabilità orizzontale. Se più policy impongono alla destinazione scalabile un dimensionamento orizzontale o verticale contemporaneamente, viene dimensionata in base al criterio che fornisce la capacità massima sia per riduzione orizzontale che per il dimensionamento orizzontale.

2. Metriche personalizzate per il monitoraggio degli obiettivi: prestare attenzione quando si utilizzano le metriche personalizzati per il monitoraggio degli obiettivi, poiché il dimensionamento automatico è più adatto alla scalabilità orizzontale/ridimensionamento proporzionale alle modifiche delle metriche scelte per la policy. Se tali metriche non cambiano proporzionalmente alle operazioni di dimensionamento utilizzate per la creazione della policy, potrebbero verificarsi operazioni di scalabilità orizzontale o ridimensionamento continue che possono influire sulla disponibilità o sui costi.

Per i cluster dei livelli di dati (tipi di istanza della famiglia r6gd), evita di utilizzare metriche basate sulla memoria per il dimensionamento.

3. Dimensionamento pianificato: se determini che il carico di lavoro è deterministico (aumenta o diminuisce in un momento specifico), ti consigliamo di utilizzare il dimensionamento pianificato e configurare la capacità obiettivo in base alle esigenze. Il monitoraggio dell'obiettivo è ideale per carichi di lavoro non deterministici e per il cluster per un funzionamento con la metrica obiettivo richiesta, con aumento orizzontale quando occorrono più risorse e riduzione orizzontale quando occorrono meno risorse.
4. Disabilita riduzione orizzontale: il dimensionamento automatico nel tracciamento obiettivi è ideale per cluster con aumento e riduzione graduale dei carichi di lavoro, in quanto picchi e gole nelle metriche possono generare oscillazioni di aumento e riduzione orizzontale consecutive. Per evitare tali oscillazioni, puoi iniziare con la riduzione orizzontale disabilitata e utilizzarla manualmente in un secondo momento in base alla necessità.
5. Test dell'applicazione: è preferibile testare l'applicazione con i carichi di lavoro minimi e massimi stimati per determinare il numero minimo e massimo assoluto di partizioni/repliche necessarie al cluster durante la creazione policy di dimensionamento per evitare problemi di disponibilità. La scalabilità automatica può effettuare l'aumento orizzontale fino alla soglia massima e la riduzione orizzontale fino alla soglia minima configurata per l'obiettivo.
6. Definizione del valore target: è possibile analizzare le CloudWatch metriche corrispondenti per l'utilizzo del cluster in un periodo di quattro settimane per determinare la soglia del valore target. Se non sei ancora certo del valore da scegliere, ti consigliamo di iniziare con il valore predefinito minimo supportato della metrica.
7. AutoScaling on Target Tracking è più adatto per i cluster con distribuzione uniforme dei carichi di lavoro tra la dimensione degli shard/repliche. Avere una distribuzione non uniforme può portare a:
 - Dimensionamento quando non richiesto a causa di picco/dip del carico di lavoro su alcuni hot partizioni/repliche.
 - Non dimensionamento quando richiesto a causa della media complessiva vicina alla destinazione anche se si dispone di partizioni/repliche hot.

Note

Quando si ridimensiona il cluster, ElastiCache replicherà automaticamente le funzioni caricate in uno dei nodi esistenti (selezionati a caso) sui nuovi nodi. Se il cluster dispone di Redis 7.0 o versioni successive e l'applicazione utilizza [Funzioni Redis](#), consigliamo di caricare tutte le

funzioni in tutte le partizioni prima di aumentare orizzontalmente il cluster in modo che non contenga funzioni diverse su partizioni differenti.

Dopo la registrazione a AutoScaling, tieni presente quanto segue:

- Esistono limitazioni sulle configurazioni del dimensionamento automatico supportate, per cui è preferibile non modificare la configurazione di un gruppo di replica registrato per il dimensionamento automatico. Di seguito vengono mostrati gli esempi:
 - Modifica manuale del tipo di istanza in tipi non supportati.
 - Associazione del gruppo di replica a un Global Datastore.
 - Modifica `ReservedMemoryPercent` parametro .
 - Aumento/riduzione manuale di partizioni/repliche oltre la capacità minima/massima configurata durante la creazione delle policy.

Utilizzo di Auto Scaling con i shard

Di seguito vengono fornite informazioni dettagliate sul monitoraggio degli obiettivi e sui criteri pianificati e su come applicarli utilizzando la AWS Management Console AWS CLI e API.

Policy di dimensionamento con monitoraggio degli obiettivi

Con le policy di dimensionamento con monitoraggio degli obiettivi, puoi scegliere un parametro e impostare un valore obiettivo. ElastiCache for Redis Auto Scaling crea e gestisce gli allarmi di CloudWatch che attivano la policy di dimensionamento e calcola le regolazioni di dimensionamento in base al parametro e al valore obiettivo. La policy di dimensionamento aggiunge o rimuove partizioni in base alle necessità, per mantenere il parametro al valore di destinazione specificato o vicino a esso. Oltre a mantenere il parametro vicino al valore di destinazione, una policy di dimensionamento di monitoraggio dei target si adatta anche alle fluttuazioni del parametro dovute a un modello di carico fluttuante e riduce al minimo le fluttuazioni rapide nella capacità del parco istanze.

Ad esempio, si consideri una policy di dimensionamento che utilizza la media predefinita `ElastiCachePrimaryEngineCPUUtilizationMetrica` con il valore di destinazione configurato. Tale policy può mantenere l'utilizzo della CPU al valore di destinazione specificato o vicino al valore di destinazione specificato.

Metriche predefinite

Una metrica predefinita è una struttura che fa riferimento a un nome, una dimensione e una statistica (average) specifici di una determinata metrica CloudWatch. La policy di dimensionamento automatico definisce le seguenti metriche predefinite per il cluster:

Nome parametro predefinito	Nome della metrica CloudWatch	Dimensione della metrica CloudWatch	Tipi di istanza non idonei
ElastiCachePrimaryEngineCPUUtilization	EngineCPUUtilization	Replicati onGroupId, Ruolo = Primario	Nessuno
ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage	DatabaseCapacityUsageCountedForEvictPercentage	Metriche del gruppo di replica Redis	Nessuno
ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage	DatabaseMemoryUsageCountedForEvictPercentage	Metriche del gruppo di replica Redis	R6gd

I tipi di istanza dei livelli di dati non possono utilizzare

ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage, poiché questi tipi di istanza archiviano i dati in memoria e in SSD. Il caso d'uso previsto per le istanze dei livelli di dati è avere il 100% di utilizzo della memoria e riempire l'SSD in base alle esigenze.

Criteri Auto Scaling per le partizioni

Quando il servizio rileva che la metrica predefinita è uguale o maggiore dell'impostazione Target, la capacità delle partizioni verrà aumentata automaticamente. ElastiCache for Redis ridimensiona i frammenti del cluster di un conteggio uguale al più grande dei due numeri: Variazione percentuale da Target e 20% dei frammenti correnti. Per la scalabilità in, ElastiCache for Redis non verrà automaticamente scale-in a meno che il valore complessivo della metrica non sia inferiore al 75% della destinazione definita.

Per un esempio di scala orizzontale, se hai 50 frammenti e

- Se il tuo Target viene violato del 30%, ElastiCache for Redis si riduce del 30%, il che si traduce in 65 frammenti per cluster.
- se il tuo Target viola del 10%, ElastiCache for Redis ridimensiona per impostazione predefinita minimo del 20%, il che si traduce in 60 frammenti per cluster.

Per un esempio di scalabilità, se è stato selezionato un valore Target pari al 60%, ElastiCache for Redis non verrà scalata automaticamente fino a quando la metrica non sarà inferiore o uguale al 45% (25% al di sotto del target 60%).

Considerazioni su Auto Scaling

Tieni a mente le seguenti considerazioni:

- Una policy di dimensionamento di monitoraggio obiettivi presuppone che essa debba eseguire un dimensionamento orizzontale quando il parametro specificato supera il valore di destinazione. Non puoi utilizzare una policy di dimensionamento di monitoraggio obiettivi per il dimensionamento orizzontale quando il parametro specificato è inferiore al valore di destinazione. ElastiCache for Redis ridimensiona i frammenti di una deviazione minima del 20% del target dei frammenti esistenti nel cluster.
- Una policy di dimensionamento di monitoraggio obiettivi non esegue il dimensionamento quando il parametro specificato non dispone di dati sufficienti. Non esegue la riduzione orizzontale perché l'insufficienza di dati non viene interpretata come basso utilizzo.
- Potrebbero esserci delle differenze tra il valore di destinazione e i punti di dati dei parametri reali. Ciò avviene perché Auto Scaling di ElastiCache for Redis agisce sempre con prudenza, arrotondando per eccesso o per difetto quando determina la capacità da aggiungere o rimuovere. In questo modo si impedisce l'aggiunta di capacità insufficiente o la rimozione di capacità eccessiva.

- Per garantire la disponibilità delle applicazioni, il servizio aumenta in proporzione al parametro il più veloce possibile, ma si riduce in modo più conservativo.
- È possibile avere più policy di dimensionamento di monitoraggio obiettivi per un cluster ElastiCache for Redis purché ciascuna di esse utilizzi un parametro diverso. Lo scopo di Auto Scaling per ElastiCache for Redis è sempre quello di assegnare la priorità alla disponibilità, quindi il suo comportamento varia a seconda che le policy di monitoraggio degli obiettivi siano pronte o meno per l'aumento o la riduzione orizzontale. Il servizio viene aumentato se una qualsiasi delle policy di monitoraggio obiettivi è pronta per l'aumento, ma viene ridotto solo se tutte le policy di monitoraggio obiettivi (con la porzione di riduzione abilitata) sono pronte per la riduzione.
- Non modificare o eliminare gli allarmi di CloudWatch che ElastiCache for Redis Auto Scaling gestisce per una policy di dimensionamento con monitoraggio degli obiettivi. ElastiCache for Redis Auto Scaling elimina gli allarmi automaticamente quando elimini la policy di dimensionamento.
- ElastiCache for Redis Auto Scaling non ti impedisce di modificare manualmente le partizioni del cluster. Queste regolazioni manuali non influiscono sugli allarmi CloudWatch esistenti collegati alla politica di ridimensionamento, ma possono influire sulle metriche che potrebbero attivare questi allarmi CloudWatch.
- Questi allarmi CloudWatch gestiti da Auto Scaling sono definiti sulla metrica AVG in tutti le partizioni del cluster. Quindi, avere frammenti caldi può comportare uno scenario di:
 - dimensionamento quando non richiesto a causa del carico su alcuni frammenti caldi che attivano un allarme CloudWatch
 - non dimensionare quando richiesto a causa di AVG aggregato su tutti le partizioni che influiscono sull'allarme non violato.
- Si applicano ancora i limiti di default ElastiCache for Redis sui nodi per cluster. Pertanto, quando si opta per il Auto Scaling e se si prevede che i nodi massimi siano superiori al limite di default, è necessario richiedere un aumento del limite a [AWS Restrizioni dei servizi](#) e scegliere il tipo di limiteNodi per cluster per tipo di istanza.
- Assicurarsi di disporre di sufficienti ENI (Elastic Network Interfaces) disponibili nel VPC, necessarie durante la scalabilità orizzontale. Per ulteriori informazioni, consulta [Interfacce di rete elastiche](#).
- Se la capacità disponibile da EC2 è insufficiente, ElastiCache per Redis Auto Scaling non esegue il dimensionamento e differisce l'operazione quando la capacità è disponibile.
- ElastiCache for Redis Auto Scaling durante la scale-in non rimuoverà i frammenti con slot aventi una dimensione dell'elemento superiore a 256 MB post-serializzazione.
- Durante la scale-in non rimuoverà i frammenti se la memoria disponibile è insufficiente nella configurazione di partizioni risultante.

Aggiunta di una policy di dimensionamento

Puoi aggiungere una policy di dimensionamento usando la AWS Management Console.

Per aggiungere una policy di Auto Scaling a un cluster ElastiCache for Redis

1. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione, scegli Redis.
3. Scegli il cluster a cui aggiungere una policy (seleziona il nome del cluster e non il pulsante a sinistra del cluster).
4. Seleziona la scheda Policy di Auto Scaling.
5. Scegli add dynamic scaling (Aggiungi dimensionamento automatico).
6. In Nome policy, immetti un nome per la policy.
7. Per Dimensione scalabile scegli partizioni.
8. Per il parametro di destinazione, scegli uno dei seguenti modi:
 - Utilizzo primario della CPU per creare una policy basata sull'utilizzo medio della CPU.
 - Memoria per creare una policy basata sulla memoria media del database.
 - Capacità per creare una policy basata sull'utilizzo della capacità database media. La metrica Capacità include l'utilizzo della memoria e di SSD per le istanze dei livelli di dati e l'utilizzo della memoria per tutti gli altri tipi di istanza.
9. Per il valore target, scegli un valore maggiore o uguale a 35 e inferiore o uguale a 70. Il dimensionamento automatico manterrà questo valore per la metrica target selezionata su tutte le partizioni ElastiCache:
 - Utilizzo della CPU primaria: mantiene il valore target per la metrica `EngineCPUUtilization` sui nodi primari.
 - Memoria: mantiene il valore target per la metrica `DatabaseMemoryUsageCountedForEvictPercentage`
 - Capacità mantiene il valore target per la metrica `DatabaseCapacityUsageCountedForEvictPercentage`,

Le partizioni del cluster vengono aggiunte o rimosse per tenere il parametro vicino al valore specificato.

10. (Facoltativo) I periodi di attesa di riduzione o aumento non sono supportati dalla console. Utilizzo della AWS CLI per modificare i valori del tempo di recupero.
11. Per Capacità minima, digita il numero minimo di repliche ElastiCache che la policy di Auto Scaling deve mantenere.
12. Per Capacità massima, digita il numero massimo di repliche che la policy di Auto Scaling di ElastiCache for Redis deve mantenere. Questo valore deve essere maggiore o uguale a 250.
13. Seleziona Crea.

Registrazione di un obiettivo scalabile

Prima di poter utilizzare Auto Scaling con un cluster ElastiCache for Redis, registrare il cluster con ElastiCache per il ridimensionamento automatico Redis. Questa operazione consente di definire la dimensione e i limiti di dimensionamento da applicare al cluster. La scalabilità automatica ElastiCache for Redis ridimensiona dinamicamente il cluster ElastiCache for Redis lungo il `elasticache:replication-group:NodeGroups` dimensione scalabile, che rappresenta il numero di frammenti di cluster.

Utilizzando la AWS CLI

Per registrare il cluster ElastiCache for Redis, utilizzare il comando [registra target scalabile](#) con i parametri seguenti.

- `--service-namespace` – Imposta questo valore su `elasticache`
- `--resource-id` – L'identificatore della risorsa per il cluster ElastiCache for Redis. Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster database ElastiCache, ad esempio `replication-group/myscalablecluster`.
- `--scalable-dimension`: imposta questo valore su `elasticache:replication-group:NodeGroups`.
- `--max-capacity` — Il numero massimo di partizioni che deve essere gestito da ElastiCache per il dimensionamento automatico Redis. Per informazioni sulla relazione tra `--min-capacity`, `--max-capacity` e il numero di partizioni nel cluster, consulta [Capacità minima e massima](#).
- `--min-capacity` — Il numero minimo di partizioni che deve essere gestito da ElastiCache per la scalabilità automatica Redis. Per informazioni sulla relazione tra `--min-capacity`, `--max-capacity` e il numero di partizioni nel cluster, consulta [Capacità minima e massima](#).

Example

Nell'esempio seguente viene registrato un cluster ElastiCache for Redis denominato `myscalablecluster`. La registrazione indica che il cluster deve essere dimensionato dinamicamente per avere da uno a dieci partizioni .

Per Linux, macOS o Unix:

```
aws application-autoscaling register-scalable-target \  
  --service-namespace elasticache \  
  --resource-id replication-group/myscalablecluster \  
  --scalable-dimension elasticache:replication-group:NodeGroups \  
  --min-capacity 1 \  
  --max-capacity 10 \  

```

Per Windows:

```
aws application-autoscaling register-scalable-target ^  
  --service-namespace elasticache ^  
  --resource-id replication-group/myscalablecluster ^  
  --scalable-dimension elasticache:replication-group:NodeGroups ^  
  --min-capacity 1 ^  
  --max-capacity 10 ^  

```

Utilizzo dell'API

Per registrare il cluster ElastiCache utilizza il comando [register-scalable-target](#) con i parametri seguenti:

- `ServiceNamespace` : Impostare questo valore su `elasticache`.
- `ResourceID`: l'identificatore della risorsa per il cluster ElastiCache. Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster ElastiCache for Redis, ad esempio `replication-group/myscalablecluster`.
- `ScalableDimension` : Impostare questo valore su `elasticache:replication-group:NodeGroups`.
- `MinCapacity` — Il numero minimo di partizioni che deve essere gestito dalla scalabilità automatica ElastiCache for Redis. Per informazioni sulla relazione tra `—capacità-min`, `—capacità-max` il numero di repliche nel cluster, consulta [Capacità minima e massima](#).

- **CapacitàMax** — Il numero massimo di partizioni che deve essere gestito dalla scalabilità automatica ElastiCache for Redis. Per informazioni sulla relazione tra `—capacità-min`, `—capacità-max` il numero di repliche nel cluster, consulta [Capacità minima e massima](#).

Example

Nell'esempio seguente viene registrato un cluster ElastiCache for Redis denominato `myscalecluster` con l'API Application Auto Scaling. Questa registrazione indica che il cluster database deve essere dimensionato dinamicamente per avere da uno a 5 repliche .

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalecluster",
  "ScalableDimension": "elasticache:replication-group:NodeGroups",
  "MinCapacity": 1,
  "MaxCapacity": 5
}
```

Definizione di una policy di dimensionamento

Una configurazione della policy di dimensionamento di monitoraggio degli obiettivi è rappresentata da un blocco JSON in cui sono definiti i valori dei parametri e della destinazione. Puoi salvare una configurazione della policy di dimensionamento come un blocco JSON in un file di testo. Si utilizza quel file di testo quando si richiama l'API AWS CLI Application Auto Scaling. Per ulteriori informazioni sulla sintassi di configurazione delle policy, consulta [TargetTrackingScalingPolicyConfiguration](#) l'Application Auto Scaling API Reference.

Le seguenti opzioni sono disponibili per definire una configurazione di una policy di dimensionamento con monitoraggio degli obiettivi.

Argomenti

- [Utilizzo di un parametro di default](#)
- [Utilizzo di un parametro personalizzato](#)
- [Utilizzo di periodi di attesa](#)
- [Disabilitazione dell'attività di riduzione orizzontale](#)
- [Applicazione di una policy di dimensionamento](#)

Utilizzo di un parametro di default

Utilizzando metriche predefinite, puoi definire rapidamente una politica di scalabilità del tracciamento del target per un cluster for Redis che funziona con il tracciamento del target in Redis Auto Scaling. ElastiCache ElastiCache

Attualmente, ElastiCache per Redis supporta le seguenti metriche predefinite in Redis Auto ElastiCache NodeGroup Scaling:

- `ElastiCachePrimaryEngineUtilizzo` della CPU: il valore medio della `EngineCPUUtilization` metrica in CloudWatch tutti i nodi primari del cluster for Redis. ElastiCache
- `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`— Il valore medio della `DatabaseMemoryUsageCountedForEvictPercentage` metrica in tutti i nodi primari del CloudWatch cluster for Redis. ElastiCache
- `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage`— Il valore medio della `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` metrica in tutti i nodi primari del CloudWatch cluster ElastiCache for Redis.

Per ulteriori informazioni sulle metriche `EngineCPUUtilization`, `DatabaseMemoryUsageCountedForEvictPercentage` e `DatabaseCapacityUsageCountedForEvictPercentage`, consultare [Monitoraggio dell'utilizzo con i parametri di CloudWatch](#). Per utilizzare un parametro di default nella policy di dimensionamento, crea una configurazione di monitoraggio degli obiettivi per la policy di dimensionamento. Questa configurazione deve includere una `PredefinedMetricSpecification` per la metrica predefinita e una `TargetValue` per il valore di destinazione di quella metrica.

Example

L'esempio seguente descrive una tipica configurazione delle policy per il ridimensionamento del target-tracking per un cluster for Redis. ElastiCache In questa configurazione, la metrica

ElastiCachePrimaryEngineCPUUtilization predefinita viene utilizzata ElastiCache per regolare il cluster Redis in base a un utilizzo medio della CPU del 40% su tutti i nodi primari del cluster.

```
{
  "TargetValue": 40.0,
  "PredefinedMetricSpecification":
  {
    "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
  }
}
```

Utilizzo di un parametro personalizzato

Utilizzando dei parametri personalizzati, è possibile definire una policy di dimensionamento di monitoraggio degli obiettivi che soddisfi i requisiti personalizzati. È possibile definire una metrica personalizzata basata su qualsiasi ElastiCache metrica che cambia in proporzione alla scalabilità. Non tutte le ElastiCache metriche funzionano per il tracciamento degli obiettivi. Il parametro deve essere un parametro di utilizzo valido e deve descrivere quanto è impegnata un'istanza. Il valore del parametro deve aumentare o diminuire in proporzione al numero di partizioni nel cluster. Questo aumento o riduzione proporzionale è necessario per utilizzare i dati del parametro per aumentare orizzontalmente in modo proporzionale il numero di repliche .

Example

Il seguente esempio descrive una configurazione di monitoraggio degli obiettivi per una policy di dimensionamento. In questa configurazione, una metrica personalizzata regola un ElastiCache cluster Redis in base a un utilizzo medio della CPU del 50% su tutti gli shard di un cluster denominato `my-db-cluster`

```
{
  "TargetValue": 50,
  "CustomizedMetricSpecification":
  {
    "MetricName": "EngineCPUUtilization",
    "Namespace": "AWS/ElastiCache",
    "Dimensions": [
      {
        "Name": "RelicationGroup", "Value": "my-db-cluster"
      }
    ],
  },
}
```

```
    {
      "Name": "Role","Value": "PRIMARY"
    }
  ],
  "Statistic": "Average",
  "Unit": "Percent"
}
```

Utilizzo di periodi di attesa

È possibile specificare un valore, in secondi, per `ScaleOutCooldown` per aggiungere un periodo di attesa per il dimensionamento orizzontale del cluster. Allo stesso modo, è possibile aggiungere un valore, in secondi, per `ScaleInCooldown` per aggiungere un periodo di attesa per la riduzione del cluster. Per ulteriori informazioni, consulta [TargetTrackingScalingPolicyConfiguration](#) l'Application Auto Scaling API Reference.

Il seguente esempio descrive una configurazione di monitoraggio degli obiettivi per una policy di dimensionamento. In questa configurazione, la metrica `ElastiCachePrimaryEngineCPUUtilization` predefinita viene utilizzata per regolare un cluster ElastiCache per Redis in base a un utilizzo medio della CPU del 40% su tutti i nodi primari di quel cluster. La configurazione fornisce un tempo di raffreddamento di riduzione orizzontale di 10 minuti e un tempo di raffreddamento di aumento di 5 minuti.

```
{
  "TargetValue": 40.0,
  "PredefinedMetricSpecification":
  {
    "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
  },
  "ScaleInCooldown": 600,
  "ScaleOutCooldown": 300
}
```

Disabilitazione dell'attività di riduzione orizzontale

È possibile impedire che la configurazione della policy di scalabilità target-tracking venga scalata nel cluster for Redis disabilitando l'attività di scalabilità. ElastiCache La disabilitazione delle attività di riduzione orizzontale impedisce alla policy di dimensionamento di eliminare le partizioni, consentendo tuttavia alla policy di dimensionamento di crearle in base alle esigenze.

È possibile specificare un valore booleano per `DisableScaleIn` per abilitare o disabilitare l'attività di riduzione orizzontale per il cluster. Per ulteriori informazioni, consulta [TargetTrackingScalingPolicyConfiguration](#) l'Application Auto Scaling API Reference.

Il seguente esempio descrive una configurazione di monitoraggio degli obiettivi per una policy di dimensionamento. In questa configurazione, la metrica `ElastiCachePrimaryEngineCPUUtilization` predefinita regola un cluster Redis in base a un ElastiCache utilizzo medio della CPU del 40% su tutti i nodi primari di quel cluster. La configurazione disabilita l'attività di riduzione per la policy di dimensionamento.

```
{
  "TargetValue": 40.0,
  "PredefinedMetricSpecification":
  {
    "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
  },
  "DisableScaleIn": true
}
```

Applicazione di una policy di dimensionamento

Dopo aver registrato il cluster con ElastiCache for Redis auto scaling e aver definito una policy di scaling, applichi la policy di scaling al cluster registrato. Per applicare una politica di scalabilità a un cluster ElastiCache for Redis, puoi utilizzare AWS CLI o l'API Application Auto Scaling.

Applicazione di una politica di scalabilità utilizzando il AWS CLI

Per applicare una politica di scalabilità al cluster ElastiCache for Redis, utilizza il [put-scaling-policy](#) comando con i seguenti parametri:

- `—policy-name—` Il nome della policy di dimensionamento.
- `—policy-type—` Impostare questo valore su `TargetTrackingScaling`.
- `--resource-id` — L'identificatore di risorsa per for Redis. ElastiCache Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster for Redis, ad esempio. `ElastiCache replication-group/myscalablecluster`
- `—service-namespace—` Impostare questo valore su `elasticsearch`.
- `—scalable-dimension—` Impostare questo valore su `elasticsearch:replication-group:NodeGroups`.

- `--target-tracking-scaling-policy-configuration` — La configurazione della policy di scalabilità target-tracking da utilizzare per il cluster for Redis. ElastiCache

Nell'esempio seguente, si applica una politica di scalabilità di tracciamento della destinazione denominata `myscalablepolicy` a un cluster for Redis denominato `with ElastiCache for Redis auto myscalablecluster scaling`. ElastiCache Per fare ciò, usa la configurazione della policy salvata in un file denominato `config.json`.

Per Linux, macOS o Unix:

```
aws application-autoscaling put-scaling-policy \  
  --policy-name myscalablepolicy \  
  --policy-type TargetTrackingScaling \  
  --resource-id replication-group/myscalablecluster \  
  --service-namespace elasticache \  
  --scalable-dimension elasticache:replication-group:NodeGroups \  
  --target-tracking-scaling-policy-configuration file://config.json
```

Per Windows:

```
aws application-autoscaling put-scaling-policy ^  
  --policy-name myscalablepolicy ^  
  --policy-type TargetTrackingScaling ^  
  --resource-id replication-group/myscalablecluster ^  
  --service-namespace elasticache ^  
  --scalable-dimension elasticache:replication-group:NodeGroups ^  
  --target-tracking-scaling-policy-configuration file://config.json
```

Applicazione di una policy di dimensionamento con l'API

Per applicare una politica di scalabilità al cluster ElastiCache for Redis, utilizza il [PutScalingPolicy](#) AWS CLI comando con i seguenti parametri:

- `—policy-name`— Il nome della policy di dimensionamento.
- `--resource-id` — L'identificatore di risorsa per for Redis. ElastiCache Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster for Redis, ad esempio. ElastiCache `replication-group/myscalablecluster`

- `—service-namespace—` Impostare questo valore su `elasticsearch`.
- `—scalable-dimension—` Impostare questo valore su `elasticsearch:replication-group:NodeGroups`.
- `-- target-tracking-scaling-policy -configuration` — La configurazione della policy di scalabilità `target-tracking` da utilizzare per il cluster for Redis. ElastiCache

Nell'esempio seguente, si applica una politica di scalabilità di tracciamento della destinazione denominata `myscalablepolicy` a un cluster for Redis denominato `with ElastiCache for Redis auto myscalesablecluster scaling`. ElastiCache Si utilizza una policy di configurazione in base al parametro di default `ElastiCachePrimaryEngineCPUUtilization`.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.PutScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "PolicyName": "myscalablepolicy",
  "ServiceNamespace": "elasticsearch",
  "ResourceId": "replication-group/myscalesablecluster",
  "ScalableDimension": "elasticsearch:replication-group:NodeGroups",
  "PolicyType": "TargetTrackingScaling",
  "TargetTrackingScalingPolicyConfiguration": {
    "TargetValue": 40.0,
    "PredefinedMetricSpecification":
    {
      "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
    }
  }
}
```

Modifica di una policy di dimensionamento

È possibile modificare una policy di dimensionamento con la AWS Management Console, AWS CLI o l'API di Application Auto Scaling.

Modifica di una policy di dimensionamento usando la AWS Management Console

Per modificare una policy di Auto Scaling per un cluster ElastiCache for Redis

1. Accedere alla AWS Management Console e aprire la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione, scegli Redis.
3. Scegli il cluster a cui aggiungere una policy (seleziona il nome del cluster e non il pulsante a sinistra del cluster).
4. Seleziona la scheda Policy di Auto Scaling.
5. In Policy di dimensionamento scegli il pulsante a sinistra della policy di Auto Scaling su cui eseguire la modifica, quindi scegli Modifica (Modifica).
6. Apportare le modifiche necessarie alla policy.
7. Scegli Modifica.

Modifica di una policy di dimensionamento con AWS CLI o l'API

È possibile utilizzare la AWS CLI o l'API Application Auto Scaling per modificare una policy di dimensionamento nello stesso modo con cui si applica una policy di dimensionamento:

- Quando si utilizza l'AWS CLI, specificare il nome della policy da modificare nel parametro `--policy-name`. Specifica i nuovi valori per i parametri che desideri modificare.
- Quando si utilizza l'API Application Auto Scaling, specificare il nome della policy da modificare nel parametro `PolicyName`. Specifica i nuovi valori per i parametri che desideri modificare.

Per ulteriori informazioni, consulta [Applicazione di una policy di dimensionamento](#).

Eliminazione di una policy di dimensionamento

È possibile eliminare una policy di dimensionamento con la AWS Management Console, AWS CLI o l'API di Application Auto Scaling.

Eliminazione di una policy di dimensionamento usando la AWS Management Console

Per eliminare una policy di Auto Scaling per un cluster ElastiCache for Redis

1. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.

2. Nel riquadro di navigazione, scegli Redis.
3. Scegli il cluster a cui aggiungere una policy (seleziona il nome del cluster e non il pulsante a sinistra del cluster).
4. Seleziona la scheda Policy di Auto Scaling.
5. Nella sezione Scaling Policies (Policy di Scaling), scegliere la policy di Auto Scaling e successivamente Delete (Elimina).

Eliminazione di una policy di dimensionamento usando la AWS CLI

Per eliminare una policy di dimensionamento nel cluster ElastiCache for Redis, utilizza il comando [elimina-scaling-policy](#) AWS CLI con i parametri seguenti.

- `—nome-policy—` Il nome della policy di dimensionamento.
- `—resource-id—` L'identificatore di risorsa per ElastiCache for Redis. Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster ElastiCache for Redis, ad esempio `replication-group/myscalablecluster`.
- `—service-namespace—` Impostare questo valore su `elasticache`.
- `—scalable dimension—` Impostare questo valore su `elasticache:replication-group:NodeGroups`.

Nell'esempio seguente, eliminare una policy di dimensionamento con monitoraggio degli obiettivi denominata `myscalablepolicy` da un cluster ElastiCache for Redis denominato `myscalablecluster`.

Per Linux, macOS o Unix:

```
aws application-autoscaling delete-scaling-policy \  
  --policy-name myscalablepolicy \  
  --resource-id replication-group/myscalablecluster \  
  --service-namespace elasticache \  
  --scalable-dimension elasticache:replication-group:NodeGroups
```

Per Windows:

```
aws application-autoscaling delete-scaling-policy ^  
  --policy-name myscalablepolicy ^  
  --resource-id replication-group/myscalablecluster ^
```

```
--service-namespace elasticache ^
--scalable-dimension elasticache:replication-group:NodeGroups
```

Eliminazione di una policy di dimensionamento usando l' API

Per eliminare una policy di dimensionamento dal cluster ElastiCache for Redis, utilizzare il comando [DeleteScalingPolicy \(elimina policy dimensionamento\)](#) AWS CLI con i parametri seguenti:

- `—policy name`— Il nome della policy di dimensionamento.
- `—resource-id`— L'identificatore di risorsa per ElastiCache for Redis. Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster ElastiCache for Redis, ad esempio `replication-group/myscalablecluster`.
- `—service-namespace`— Impostare questo valore su `elasticache`.
- `—scalable dimension`— Impostare questo valore su `elasticache:replication-group:NodeGroups`.

Nell'esempio seguente, eliminare una policy di dimensionamento con monitoraggio degli obiettivi denominata `myscalablepolicy` da un cluster ElastiCache for Redis denominato `myscalablecluster`.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.DeleteScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "PolicyName": "myscalablepolicy",
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:NodeGroups"
}
```

Usare AWS CloudFormation per policy di Auto Scaling

Questo snippet mostra come creare una policy di monitoraggio degli obiettivi applicarla a una risorsa [AWS::ElastiCache::ReplicationGroup](#) utilizzando la risorsa

[AWS::ApplicationAutoScaling::ScalableTarget](#) . Utilizza le funzioni intrinseche [Fn::Join](#) e [Ref](#) per costruire la proprietà `ResourceId` con il nome logico della risorsa `AWS::ElastiCache::ReplicationGroup` specificata nello stesso modello.

```
ScalingTarget:
  Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
  Properties:
    MaxCapacity: 3
    MinCapacity: 1
    ResourceId: !Sub replication-group/${logicalName}
    ScalableDimension: 'elasticache:replication-group:NodeGroups'
    ServiceNamespace: elasticache
    RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"

ScalingPolicy:
  Type: "AWS::ApplicationAutoScaling::ScalingPolicy"
  Properties:
    ScalingTargetId: !Ref ScalingTarget
    ServiceNamespace: elasticache
    PolicyName: testpolicy
    PolicyType: TargetTrackingScaling
    ScalableDimension: 'elasticache:replication-group:NodeGroups'
    TargetTrackingScalingPolicyConfiguration:
      PredefinedMetricSpecification:
        PredefinedMetricType: ElastiCachePrimaryEngineCPUUtilization
      TargetValue: 40
```

Dimensionamento programmato

Il dimensionamento basato su una pianificazione consente di dimensionare le applicazioni in relazione alle variazioni di domanda prevedibili. Per utilizzare il dimensionamento pianificato, è possibile creare operazioni pianificate che indicano al ElastiCache for Redis di eseguire attività di dimensionamento a orari specifici. Al momento della creazione di un'operazione pianificata, è possibile specificare un cluster ElastiCache for Redis esistente, quando l'attività di dimensionamento dovrà verificarsi, la capacità minima e la capacità massima. È possibile creare operazioni pianificate sia una tantum che ricorrenti.

È possibile creare operazioni pianificate solo per i cluster ElastiCache for Redis già esistenti. Non è possibile creare operazioni pianificate contemporaneamente alla creazione di un cluster.

Per ulteriori informazioni sulla terminologia per la creazione, la gestione ed l'eliminazione di azioni pianificate, consulta [I comandi comunemente utilizzati per creazione, la gestione ed eliminazione](#)

Per creare in base a una pianificazione ricorrente:

1. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione, scegli Redis.
3. Scegli il cluster a cui desideri aggiungere una policy.
4. Scegli Gestione delle policy di Auto Scaling dal dropdown Operazioni.
5. Seleziona la scheda Policy di Auto Scaling.
6. Nella sezione Policy di Auto Scaling viene visualizzata la finestra di dialogo Aggiungi policy di dimensionamento Scegli Dimensionamento pianificato.
7. In Nome policy, immetti un nome per la policy.
8. Per Dimensione scalabile, scegliere Partizioni.
9. Per Obiettivi delle partizioni, scegliere il valore.
10. Per Ricorrenza, scegli Ricorrente.
11. Per Frequenza, scegli il rispettivo valore.
12. Per Data di inizio e Ora di inizio, scegli il momento in cui la policy entrerà in vigore.
13. Scegli Aggiungi policy.

Per creare un'operazione pianificata una tantum

1. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione, scegli Redis.
3. Scegli il cluster a cui desideri aggiungere una policy.
4. Scegli Gestione delle policy di Auto Scaling dal dropdown Operazioni.
5. Seleziona la scheda Policy di Auto Scaling.
6. Nella sezione Policy di Auto Scaling viene visualizzata la finestra di dialogo Aggiungi policy di dimensionamento Scegli Dimensionamento pianificato.
7. In Nome policy, immetti un nome per la policy.
8. Per Dimensione scalabile, scegliere Partizioni.

9. Per Obiettivi delle partizioni, scegliere il valore.
10. Per Ricorrenza, scegli Una volta.
11. Per Data di inizio e Ora di inizio, scegli il momento in cui la policy entrerà in vigore.
12. Per Data di fine scegli la data fino a quando la policy sarà in vigore.
13. Scegli Aggiungi policy.

Per eliminare un'operazione pianificata

1. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione, scegli Redis.
3. Scegli il cluster a cui desideri aggiungere una policy.
4. Scegli Gestione delle policy di Auto Scaling dal dropdown Operazioni.
5. Scegli la scheda Policy di Auto Scaling.
6. Nella sezione Policy di Auto Scaling, scegli la policy di Auto Scaling e successivamente Elimina dalla finestra di dialogo Azioni.

Per gestire il dimensionamento pianificato tramite AWS CLI

Utilizza le seguenti API di scalabilità automatica delle applicazioni:

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Utilizza AWS CloudFormation per creare un'operazione pianificata

Questo snippet mostra come creare una policy di monitoraggio degli obiettivi e applicarla a una risorsa [AWS::ElastiCache::ReplicationGroup](#) utilizzando la risorsa [AWS::ApplicationAutoScaling::ScalableTarget](#). Utilizza le funzioni intrinseche [Fn::Join](#) e [Ref](#) per costruire la proprietà ResourceId con il nome logico della risorsa `AWS::ElastiCache::ReplicationGroup` specificata nello stesso modello.

```
ScalingTarget:
```

```
Type: 'AWS::ApplicationAutoScaling::ScalableTarget'  
Properties:  
  MaxCapacity: 3  
  MinCapacity: 1  
  ResourceId: !Sub replication-group/${logicalName}  
  ScalableDimension: 'elasticache:replication-group:NodeGroups'  
  ServiceNamespace: elasticache  
  RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-  
service-role/elasticache.application-autoscaling.amazonaws.com/  
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"  
  ScheduledActions:  
    - EndTime: '2020-12-31T12:00:00.000Z'  
      ScalableTargetAction:  
        MaxCapacity: '5'  
        MinCapacity: '2'  
        ScheduledActionName: First  
        Schedule: 'cron(0 18 * * ? *)'
```

Utilizzo di Auto Scaling con le repliche

Di seguito vengono fornite informazioni dettagliate sul monitoraggio degli obiettivi e sulle policy pianificate e su come applicarli utilizzando la AWS Management Console AWS CLI e gli API.

Policy di dimensionamento con monitoraggio degli obiettivi

Con le policy di dimensionamento con monitoraggio degli obiettivi, puoi scegliere un parametro e impostare un valore obiettivo. Il servizio di AutoScaling di Amazon ElastiCache for Redis crea e gestisce gli allarmi di CloudWatch che attivano la policy di dimensionamento e calcola le regolazioni di dimensionamento in base al parametro e al valore obiettivo. La policy di dimensionamento aggiunge o rimuove le repliche uniformemente come richiesto per mantenere il parametro al valore di destinazione specificato o vicino a esso. Oltre a mantenere il parametro vicino al valore di destinazione, una policy di dimensionamento di monitoraggio dei target si adatta anche alle fluttuazioni del parametro dovute a un modello di carico fluttuante e riduce al minimo le fluttuazioni rapide nella capacità del parco istanze.

Criteri Auto Scaling per le repliche

La policy di Auto Scaling definisce il seguente parametro di default per il cluster:

ElastiCacheReplicaEngineCPUUtilization: la soglia di utilizzo di AVG EngineCPU aggregata in tutte le repliche utilizzate da ElastiCache for Redis per attivare un'operazione di scalabilità automatica. È possibile impostare l'obiettivo di utilizzo tra il 35 percento e il 70 percento.

Quando il servizio rileva che il tuo parametro `ElastiCacheReplicaEngineCPUUtilization` è uguale o maggiore a quella dell'impostazione `Target`, aumenta automaticamente le repliche tra gli shard. ElastiCache for Redis dimensiona orizzontalmente le repliche del cluster di un conteggio uguale al più grande dei due numeri: `Variazione percentuale da Target` e una replica. Per la riduzione orizzontale, ElastiCache for Redis non effettuerà automaticamente la riduzione orizzontale a meno che il valore complessivo del parametro non sia inferiore al 75% della destinazione definita.

Per un esempio di aumento orizzontale, se si dispone di 5 shard e 1 replica ciascuno:

Se il target viene violato del 30 per cento, ElastiCache for Redis ridimensiona di 1 replica (max (0.3, default 1)) su tutti i frammenti. Il che si traduce in 5 frammenti con 2 repliche ciascuna,

Per un esempio per ridurre orizzontalmente, se è stato selezionato `Valore target` pari al 60%, ElastiCache for Redis non attiverà la scalabilità automatica per ridurre orizzontalmente fino a quando il parametro non sarà inferiore o uguale al 45% (25% al di sotto del target 60%).

Considerazioni su Auto Scaling

Tieni a mente le seguenti considerazioni:

- Una policy di dimensionamento di monitoraggio obiettivi presuppone che essa debba eseguire un dimensionamento orizzontale quando il parametro specificato supera il valore di destinazione. Non puoi utilizzare una policy di dimensionamento di monitoraggio obiettivi per il dimensionamento orizzontale quando il parametro specificato è inferiore al valore di destinazione. ElastiCache for Redis dimensiona orizzontalmente le repliche al massimo (% di deviazione arrotondata da `Target`, default 1) delle repliche esistenti su tutte le partizioni del cluster.
- Una policy di dimensionamento di monitoraggio obiettivi non esegue il dimensionamento quando il parametro specificato non dispone di dati sufficienti. Non esegue la scalabilità in quanto la carenza di dati non viene interpretata come basso utilizzo.
- Potrebbero esserci delle differenze tra il valore di destinazione e i punti di dati dei parametri reali. Ciò avviene perché Auto Scaling di ElastiCache for Redis agisce sempre con prudenza, arrotondando per eccesso o per difetto quando determina la capacità da aggiungere o rimuovere. In questo modo si impedisce l'aggiunta di capacità insufficiente o la rimozione di capacità eccessiva.
- Per garantire la disponibilità delle applicazioni, il servizio aumenta orizzontalmente in proporzione al parametro il più veloce possibile, ma riduce orizzontalmente in modo più graduale con una riduzione orizzontale di 1 replica nelle partizioni nel cluster.

- È possibile avere più policy di dimensionamento di monitoraggio obiettivi per un cluster ElastiCache for Redis purché ciascuna di esse utilizzi un parametro diverso. Lo scopo di Auto Scaling per ElastiCache for Redis è sempre quello di assegnare la priorità alla disponibilità, quindi il suo comportamento varia a seconda che le policy di monitoraggio degli obiettivi siano pronte o meno per l'aumento o la riduzione orizzontale. Il servizio viene aumentato se una qualsiasi delle policy di monitoraggio obiettivi è pronta per l'aumento, ma viene ridotto solo se tutte le policy di monitoraggio obiettivi (con la porzione di riduzione abilitata) sono pronte per la riduzione.
- Non modificare o eliminare gli allarmi di CloudWatch che ElastiCache for Redis Auto Scaling gestisce per una policy di dimensionamento con monitoraggio degli obiettivi. ElastiCache for Redis Auto Scaling elimina gli allarmi automaticamente quando elimini la policy di dimensionamento o elimini il cluster.
- ElastiCache for Redis Auto Scaling non ti impedisce di modificare manualmente le repliche tra shard. Queste regolazioni manuali non influiscono sugli allarmi CloudWatch esistenti collegati alla politica di ridimensionamento, ma possono influire sulle metriche che potrebbero attivare questi allarmi CloudWatch.
- Questi allarmi CloudWatch gestiti da Auto Scaling sono definiti sulla metrica AVG in tutti le partizioni del cluster. Quindi, avere frammenti caldi può comportare uno scenario di:
 - dimensionamento quando non richiesto a causa del carico su alcuni frammenti caldi che attivano un allarme CloudWatch
 - non dimensionare quando richiesto a causa di AVG aggregato su tutti le partizioni che influiscono sull'allarme non violato.
- Si applicano ancora i limiti di default ElastiCache for Redis sui nodi per cluster. Pertanto, quando si opta per il Auto Scaling e se si prevede che i nodi massimi siano superiori al limite di default, è necessario richiedere un aumento del limite a [AWS Restrizioni dei servizi](#) e scegliere il tipo di limiteNodi per cluster per tipo di istanza.
- Assicurarsi di disporre di sufficienti ENI (Elastic Network Interfaces) disponibili nel VPC, necessarie durante la scalabilità orizzontale. Per ulteriori informazioni, consulta [Interfacce di rete elastiche](#).
- Se non è disponibile capacità sufficiente da EC2, ElastiCache for Redis Auto Scaling non verrà ridimensionata fino a quando la capacità è disponibile o se si modifica manualmente il cluster per i tipi di istanza che hanno capacità sufficiente.
- ElastiCache for Redis Auto Scaling non supporta il ridimensionamento delle repliche con un cluster che ha `ReservedMemoryPercent` meno del 25 per cento. Per ulteriori informazioni, consulta [Gestione della memoria heap](#).

Aggiunta di una policy di dimensionamento

È possibile aggiungere una politica di scalabilità utilizzando AWS Management Console

Aggiungere una politica di scalabilità utilizzando il AWS Management Console

Per aggiungere una politica di ridimensionamento automatico a un ElastiCache per Redis

1. Accedi AWS Management Console e apri la ElastiCache console Amazon all'[indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Nel riquadro di navigazione, scegli Redis.
3. Scegli il cluster a cui aggiungere una policy (seleziona il nome del cluster e non il pulsante a sinistra del cluster).
4. Seleziona la scheda Policy di Auto Scaling.
5. Scegli Add dynamic scaling (Aggiungi dimensionamento automatico).
6. In Policy di dimensionamento, scegli Add dynamic scaling (Aggiungi dimensionamento automatico).
7. In Policy Name (Nome policy), immettere un nome per la policy.
8. Per Dimensione scalabile, scegliere Repliche dalla finestra di dialogo.
9. Come valore target, digita la percentuale media di utilizzo della CPU che desideri mantenere sulle ElastiCache repliche. Questo valore deve essere tra ≥ 35 e ≤ 70 . Le repliche dei cluster vengono aggiunte o rimosse per tenere il parametro vicino al valore specificato.
10. (Facoltativo) I periodi di attesa di riduzione o aumento non sono supportati dalla console. Utilizzate il AWS CLI per modificare i valori di raffreddamento.
11. Per Capacità minima, digita il numero minimo di repliche che la policy ElastiCache for Redis Auto Scaling deve gestire.
12. Per Capacità massima, digita il numero massimo di repliche che la policy ElastiCache for Redis Auto Scaling deve mantenere. Questo valore deve essere ≥ 5 .
13. Scegli Create (Crea).

Registrazione di un obiettivo scalabile

Puoi applicare una policy di dimensionamento basata un parametro di default o personalizzato. A tale scopo, è possibile utilizzare AWS CLI o l'API Application Auto Scaling. Il primo passo è registrare il gruppo di replica ElastiCache for Redis con ElastiCache for Redis auto scaling.

Prima di poter utilizzare ElastiCache la scalabilità automatica Redis con un cluster ElastiCache for Redis, devi registrare il cluster con for ElastiCache Redis auto scaling. Lo fai per definire la dimensione di scalabilità e i limiti da applicare a quel cluster. ElastiCache for Redis auto scaling ridimensiona dinamicamente il cluster ElastiCache for Redis lungo la dimensione `elasticache:replication-group:Replicas` scalabile, che rappresenta il numero di repliche del cluster per shard.

Utilizzo della CLI

Per registrare il ElastiCache cluster, utilizza il comando con i seguenti parametri: [register-scalable-target](#)

- `—service-namespace` — Impostare questo valore su `elasticache`.
- `--resource-id` — L'identificatore di risorsa per il cluster. ElastiCache Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster for Redis, ad esempio. `ElastiCache replication-group/myscalablecluster`
- `—scalable-dimension` — Imposta questo valore su `elasticache:replication-group:Replicas`.
- `--min-capacity` — Il numero minimo di repliche da gestire per la scalabilità automatica di ElastiCache Redis. Per informazioni sulla relazione tra `—capacità-min`, `—capacità-max` il numero di repliche nel cluster, consulta [Capacità minima e massima](#).
- `--max-capacity` — Il numero massimo di repliche da gestire per la scalabilità automatica di ElastiCache Redis. Per informazioni sulla relazione tra `—capacità-min`, `—capacità-max` il numero di repliche nel cluster, consulta [Capacità minima e massima](#).

Example

Nell'esempio seguente, si registra un cluster for Redis denominato. `ElastiCache myscalablecluster` La registrazione indica che il cluster deve essere dimensionato dinamicamente per avere da una a 5 repliche .

Per Linux, macOS o Unix:

```
aws application-autoscaling register-scalable-target \  
  --service-namespace elasticache \  
  --resource-id replication-group/myscalablecluster \  
  --scalable-dimension elasticache:replication-group:Replicas \  
  --min-capacity 1 \  
  --max-capacity 5
```

```
--max-capacity 5 \
```

Per Windows:

```
aws application-autoscaling register-scalable-target ^
  --service-namespace elasticache ^
  --resource-id replication-group/myscalablecluster ^
  --scalable-dimension elasticache:replication-group:Replicas ^
  --min-capacity 1 ^
  --max-capacity 5 ^
```

Utilizzo dell'API

Per registrare il ElastiCache cluster, utilizzate il [register-scalable-target](#) comando con i seguenti parametri:

- **ServiceNamespace** — Imposta questo valore su `elasticache`.
- **resourceID**: l'identificatore della risorsa per il cluster. ElastiCache Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster ElastiCache for Redis, ad esempio. `replication-group/myscalablecluster`
- **ScalableDimension** — Imposta questo valore su. `elasticache:replication-group:Replicas`
- **MinCapacity** — Il numero minimo di repliche da ElastiCache gestire per la scalabilità automatica di Redis. Per informazioni sulla relazione tra `—capacità-min`, `—capacità-max` il numero di repliche nel cluster, consulta. [Capacità minima e massima](#).
- **MaxCapacity** — Il numero massimo di repliche da ElastiCache gestire per la scalabilità automatica di Redis. Per informazioni sulla relazione tra `—capacità-min`, `—capacità-max` il numero di repliche nel cluster, consulta. [Capacità minima e massima](#).

Example

Nell'esempio seguente, si registra un cluster ElastiCache for Redis denominato `myscalablecluster` con l'API Application Auto Scaling. Questa registrazione indica che il cluster database deve essere dimensionato dinamicamente per avere da uno a 5 repliche .

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
```

```
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:Replicas",
  "MinCapacity": 1,
  "MaxCapacity": 5
}
```

Definizione di una policy di dimensionamento

Una configurazione della policy di dimensionamento di monitoraggio degli obiettivi è rappresentata da un blocco JSON in cui sono definiti i valori dei parametri e della destinazione. Puoi salvare una configurazione della policy di dimensionamento come un blocco JSON in un file di testo. Questo file di testo viene utilizzato quando si chiama la AWS CLI o l'API Application Auto Scaling. Per ulteriori informazioni sulla sintassi della configurazione della policy, consulta [TargetTrackingScalingPolicyConfiguration](#) nella Documentazione di riferimento dell'API Application Auto Scaling

Le seguenti opzioni sono disponibili per definire una configurazione di una policy di dimensionamento con monitoraggio degli obiettivi.

Argomenti

- [Utilizzo di un parametro di default](#)
- [Modifica di una policy di dimensionamento](#)
- [Eliminazione di una policy di dimensionamento](#)
- [Usare AWS CloudFormation per policy di Auto Scaling](#)
- [Dimensionamento programmato](#)

Utilizzo di un parametro di default

Una configurazione della policy di dimensionamento di monitoraggio degli obiettivi è rappresentata da un blocco JSON in cui sono definiti i valori dei parametri e della destinazione. Puoi salvare una configurazione della policy di dimensionamento come un blocco JSON in un file di testo. Questo file di testo viene utilizzato quando si chiama la AWS CLI o l'API Application Auto

Scaling. Per ulteriori informazioni sulla sintassi della configurazione della policy, consulta [TargetTrackingScalingPolicyConfiguration](#) nella Documentazione di riferimento dell'API Application Auto Scaling

Le seguenti opzioni sono disponibili per definire una configurazione di una policy di dimensionamento con monitoraggio degli obiettivi.

Argomenti

- [Utilizzo di un parametro di default](#)
- [Utilizzo di un parametro personalizzato](#)
- [Utilizzo di periodi di attesa](#)
- [Disabilitazione dell'attività di riduzione orizzontale](#)
- [Applicazione di una policy di dimensionamento a un cluster ElastiCache for Redis](#)

Utilizzo di un parametro di default

Utilizzando dei parametri predefiniti, si può definire rapidamente una policy di dimensionamento con monitoraggio degli obiettivi per un cluster ElastiCache for Redis che funzioni bene con il monitoraggio degli obiettivi in ElastiCache for Redis Auto Scaling. Attualmente, ElastiCache per Redis supporta la seguente metrica predefinita nel dimensionamento automatico delle repliche di ElastiCache:

`ElastiCacheReplicaEngineCPUUtilization`— Il valore medio del parametro `EngineCPUUtilization` su CloudWatch su tutte le repliche nel cluster ElastiCache for Redis Il valore medio del parametro `EngineCPUUtilization` su CloudWatch in tutte le repliche nel cluster ElastiCache for Redis. Puoi trovare il valore parametro aggregato in CloudWatch sotto `ElastiCache for Redis ReplicationGroupId`, `Role` per `ReplicationGroupId` e `Replica` ruolo richiedi.

Per utilizzare un parametro di default nella policy di dimensionamento, crea una configurazione di monitoraggio degli obiettivi per la policy di dimensionamento. La configurazione deve includere un `PredefinedMetricSpecification` per il parametro di default e un `TargetValue` per il valore di destinazione del parametro.

Utilizzo di un parametro personalizzato

Utilizzando dei parametri personalizzati, è possibile definire una policy di dimensionamento di monitoraggio degli obiettivi che soddisfi i requisiti personalizzati. È possibile definire un parametro personalizzato in base a qualsiasi parametro ElastiCache for Redis che si modifichi in proporzione al dimensionamento. Non tutti i parametri ElastiCache for Redis funzionano per il monitoraggio

degli obiettivi . Il parametro deve essere un parametro di utilizzo valido e deve descrivere quanto è impegnata un'istanza. Il valore del parametro deve aumentare o diminuire in proporzione al numero di repliche nel cluster . Questo aumento o riduzione proporzionale è necessario per utilizzare i dati del parametro per aumentare o diminuire in modo proporzionale il numero di repliche .

Example

Il seguente esempio descrive una configurazione di monitoraggio degli obiettivi per una policy di dimensionamento. In questa configurazione, un parametro personalizzato regola un cluster ElastiCache for Redis in base a un utilizzo medio della CPU del 50 per cento in tutte le repliche in un cluster denominato `my-db-cluster`.

```
{"TargetValue": 50,
  "CustomizedMetricSpecification":
  {"MetricName": "EngineCPUUtilization",
    "Namespace": "AWS/ElastiCache",
    "Dimensions": [
      {"Name": "RelicationGroup", "Value": "my-db-cluster"},
      {"Name": "Role", "Value": "REPLICA"}
    ],
    "Statistic": "Average",
    "Unit": "Percent"
  }
}
```

Utilizzo di periodi di attesa

È possibile specificare un valore, in secondi, per `ScaleOutCooldown` per aggiungere un periodo di attesa per il dimensionamento orizzontale del cluster. Allo stesso modo, è possibile aggiungere un valore, in secondi, per `ScaleInCooldown` per aggiungere un periodo di attesa per la riduzione orizzontale del cluster. Per i valori predefiniti, su `ScaleInCooldown` e `ScaleOutCooldown` consulta [argetTrackingScalingPolicyConfiguration](#) nella Documentazione di riferimento dell'API Application Auto Scaling. Il seguente esempio descrive una configurazione di monitoraggio degli obiettivi per una policy di dimensionamento. In questa configurazione, il parametro di default `ElastiCacheReplicaEngineCPUUtilization` viene utilizzato per regolare un cluster ElastiCache for Redis in base a un utilizzo medio della CPU del 40 per cento in tutte le repliche in quel cluster database . La configurazione fornisce un tempo di raffreddamento di riduzione orizzontale di 10 minuti e un tempo di raffreddamento di aumento di 5 minuti.

```
{"TargetValue": 40.0,
```

```
"PredefinedMetricSpecification":  
{"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"  
},  
"ScaleInCooldown": 600,  
"ScaleOutCooldown": 300  
}
```

Disabilitazione dell'attività di riduzione orizzontale

Puoi evitare che la configurazione della policy di dimensionamento con monitoraggio degli obiettivi effettui la riduzione orizzontale del tuo cluster ElastiCache for Redis disabilitando l'attività di riduzione orizzontale. La disabilitazione delle attività di riduzione impedisce alla policy di dimensionamento di eliminare le repliche, consentendo tuttavia alla policy di dimensionamento di aggiungerle in base alle esigenze.

È possibile specificare un valore booleano per `DisableScaleIn` per abilitare o disabilitare l'attività di riduzione per il cluster database. Per maggiori informazioni su `DisableScaleIn`, consulta [TargetTrackingScalingPolicyConfiguration](#) nella Documentazione di riferimento dell'API Application Auto Scaling.

Example

Il seguente esempio descrive una configurazione di monitoraggio degli obiettivi per una policy di dimensionamento. In questa configurazione, il parametro di default `ElastiCacheReplicaEngineCPUUtilization` regola un cluster ElastiCache for Redis in base a un utilizzo medio della CPU del 40 per cento in tutte le repliche in quel cluster. La configurazione disabilita l'attività di riduzione per la policy di dimensionamento.

```
{"TargetValue": 40.0,  
  "PredefinedMetricSpecification":  
  {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"  
  },  
  "DisableScaleIn": true  
}
```

Applicazione di una policy di dimensionamento a un cluster ElastiCache for Redis

Dopo la registrazione della scalabilità automatica del cluster ElastiCache for Redis e la definizione di una policy di dimensionamento, applicare la policy di dimensionamento al cluster registrato. Per applicare una policy di dimensionamento a un cluster ElastiCache for Redis, è possibile utilizzare la AWS CLI oppure l'API Application Auto Scaling.

Utilizzando la AWS CLI

Per applicare una policy di dimensionamento al cluster ElastiCache for Redis, utilizzare il comando [put-scaling-policy](#) con i parametri seguenti:

- `--policy-name` – Il nome della policy di dimensionamento.
- `—policy-type` — Impostare questo valore su `TargetTrackingScaling`.
- `—resource-id` — L'identificatore della risorsa per il cluster ElastiCache for Redis. Per questo parametro, il tipo di risorsa è `GruppoddiReplica` e l'identificatore univoco è il nome del cluster ElastiCache for Redis, ad esempio `replication-group/myscalablecluster`.
- `—service-namespace` — Impostare questo valore su `elasticache`.
- `—scalable-dimension` — Imposta questo valore su `elasticache:replication-group:Replicas`.
- `—target-tracking-scaling-percent`— La configurazione di una policy di dimensionamento per il monitoraggio degli obiettivi da utilizzare per il cluster ElastiCache for Redis.

Example

Nell'esempio seguente si applica una policy di dimensionamento con monitoraggio degli obiettivi denominata `myscalablepolicy` a un cluster ElastiCache for Redis denominato `myscalablecluster` con la scalabilità automatica ElastiCache for Redis. Per fare ciò, usa la configurazione della policy salvata in un file denominato `config.json`.

Per Linux, macOS o Unix:

```
aws application-autoscaling put-scaling-policy \  
  --policy-name myscalablepolicy \  
  --policy-type TargetTrackingScaling \  
  --resource-id replication-group/myscalablecluster \  
  --service-namespace elasticache \  
  --scalable-dimension elasticache:replication-group:Replicas \  
  --target-tracking-scaling-policy-configuration file://config.json
```

```
{"TargetValue": 40.0,  
  "PredefinedMetricSpecification":  
  {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"  
  },
```



```
"DisableScaleIn": true
}
```

Per Windows:

```
aws application-autoscaling put-scaling-policy ^
  --policy-name myscalablepolicy ^
  --policy-type TargetTrackingScaling ^
  --resource-id replication-group/myscalablecluster ^
  --service-namespace elasticache ^
  --scalable-dimension elasticache:replication-group:Replicas ^
  --target-tracking-scaling-policy-configuration file://config.json
```

Utilizzo dell'API

Per applicare una policy di dimensionamento al cluster ElastiCache for Redis con l'API Application Auto Scaling, utilizzare l'operazione API Application Auto Scaling [PutScalingPolicy](#) con i parametri seguenti:

- **PolicyName** – Il nome della policy di dimensionamento.
- **PolicyType** — Impostare questo valore su `TargetTrackingScaling`.
- **ResourceID** : L'identificatore della risorsa per il cluster ElastiCache for Redis. Per questo parametro, il tipo di risorsa è `Gruppodireplica` e l'identificatore univoco è il nome del cluster ElastiCache for Redis, ad esempio `replication-group/myscalablecluster`.
- **ServiceNamespace** : Impostare questo valore su `elasticache`.
- **ScalableDimension** : Impostare questo valore su `elasticache:replication-group:Replicas`.
- **TargetTrackingScalingPolicyConfiguration** — La configurazione di una policy di dimensionamento con monitoraggio degli obiettivi da utilizzare per il cluster ElastiCache for Redis.

Example

Nell'esempio seguente si applica una policy di dimensionamento con monitoraggio degli obiettivi denominata `scalablepolicy` a un cluster ElastiCache for Redis denominato `myscalablecluster` con scalabilità automatica di ElastiCache for Redis. Si utilizza una policy di configurazione in base al parametro di default `ElastiCacheReplicaEngineCPUUtilization`.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.PutScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "PolicyName": "myscalablepolicy",
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:Replicas",
  "PolicyType": "TargetTrackingScaling",
  "TargetTrackingScalingPolicyConfiguration": {
    "TargetValue": 40.0,
    "PredefinedMetricSpecification":
    {
      "PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
    }
  }
}
```

Modifica di una policy di dimensionamento

È possibile modificare una policy di dimensionamento con la AWS Management Console, AWS CLI o l'API di Application Auto Scaling.

Modifica di una policy di dimensionamento usando la AWS Management Console

È possibile modificare i criteri solo con tipo Parametri predefiniti utilizzando la AWS Management Console

1. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegliere Redis
3. Scegli il cluster a cui aggiungere una policy (seleziona il nome del cluster e non il pulsante a sinistra del cluster).
4. Seleziona la scheda Policy di Auto Scaling.

5. In Policy di dimensionamento scegli il pulsante a sinistra della policy di Auto Scaling su cui eseguire la modifica, quindi scegli Modifica (Modifica).
6. Apportare le modifiche necessarie alla policy.
7. Scegli Modifica.
8. Apportare modifiche alla policy.
9. Scegli Modifica.

Modifica di una policy di dimensionamento con la AWS CLI o l'API Application Auto Scaling

È possibile utilizzare la AWS CLI o l'API Application Auto Scaling per modificare una policy di dimensionamento nello stesso modo con cui si applica una policy di dimensionamento:

- Quando si utilizza l'API Application Auto Scaling, specificare il nome della policy da modificare nel parametro `PolicyName`. Specifica i nuovi valori per i parametri che desideri modificare.

Per ulteriori informazioni, consulta [Applicazione di una policy di dimensionamento a un cluster ElastiCache for Redis](#).

Eliminazione di una policy di dimensionamento

È possibile eliminare una politica di scalabilità utilizzando AWS Management Console l' AWS CLI API Application Auto Scaling

Eliminazione di una politica di scalabilità utilizzando AWS Management Console

È possibile modificare i criteri solo con tipo Parametri predefiniti utilizzando la AWS Management Console

1. Accedi AWS Management Console e apri la ElastiCache console Amazon all'[indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Nel pannello di navigazione, scegliere Redis
3. Scegliere il cluster con la policy di Auto Scaling che si desidera eliminare.
4. Seleziona la scheda Policy Auto Scaling.
5. Nella sezione Scaling Policies (Policy di dimensionamento), scegli la policy di Auto Scaling e successivamente Delete (Elimina).

Eliminazione di una politica di scalabilità utilizzando l'API AWS CLI Application Auto Scaling

È possibile utilizzare l'API Application Auto Scaling AWS CLI o l'API Application Auto Scaling per eliminare una politica di scalabilità da un cluster. ElastiCache

CLI

Per eliminare una politica di scalabilità dal cluster ElastiCache for Redis, utilizza il [delete-scaling-policy](#) comando con i seguenti parametri:

- `--policy-name`— Il nome della policy di dimensionamento.
- `--resource-id` — L'identificatore di risorsa per il cluster for Redis. ElastiCache Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster, ad esempio. ElastiCache `replication-group/myscalablecluster`
- `--service-namespace` — Impostare questo valore su `elasticache`.
- `--scalable-dimension` — Imposta questo valore su `elasticache:replication-group:Replicas`.

Example

Nell'esempio seguente, eliminare una policy di dimensionamento con monitoraggio degli obiettivi denominata `myscalablepolicy` da un cluster ELC denominato `myscalablecluster`.

Per Linux, macOS o Unix:

```
aws application-autoscaling delete-scaling-policy \  
  --policy-name myscalablepolicy \  
  --resource-id replication-group/myscalablecluster \  
  --service-namespace elasticache \  
  --scalable-dimension elasticache:replication-group:Replicas \  

```

Per Windows:

```
aws application-autoscaling delete-scaling-policy ^ \  
  --policy-name myscalablepolicy ^ \  
  --resource-id replication-group/myscalablecluster ^ \  
  --service-namespace elasticache ^ \  
  --scalable-dimension elasticache:replication-group:Replicas ^ \  

```

API

Per eliminare una policy di scaling dal tuo cluster ElastiCache for Redis, utilizza l'operazione API [DeleteScalingPolicy](#) Application Auto Scaling con i seguenti parametri:

- **PolicyName** — Il nome della politica di scalabilità.
- **resourceID** — L'identificatore di risorsa per il cluster for Redis. ElastiCache Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster, ad esempio. `ElastiCache replication-group/myscalablecluster`
- **ServiceNamespace** — Imposta questo valore su `elasticache`.
- **ScalableDimension** — Imposta questo valore su `elasticache:replication-group:Replicas`

Nell'esempio seguente, si elimina una policy di scaling di tracciamento della destinazione denominata `myscalablepolicy` da un cluster ElastiCache for Redis denominato `myscalablecluster` con l'API Application Auto Scaling.

```
POST / HTTP/1.1
>>>>>> mainline
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.DeleteScalingPolicy
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "PolicyName": "myscalablepolicy",
  "ServiceNamespace": "elasticache",
  "ResourceId": "replication-group/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:Replicas"
}
```

Usare AWS CloudFormation per policy di Auto Scaling

Questo snippet mostra come creare un'operazione pianificata e applicarla a una risorsa [AWS::ElastiCache::ReplicationGroup](#) utilizzando la risorsa [AWS::ApplicationAutoScaling::ScalableTarget](#). Utilizza le funzioni intrinseche [Fn::Join](#) e [Ref](#) per costruire la proprietà `ResourceId` con il nome logico della risorsa `AWS::ElastiCache::ReplicationGroup` specificata nello stesso modello.

```
ScalingTarget:
  Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
  Properties:
    MaxCapacity: 0
    MinCapacity: 0
    ResourceId: !Sub replication-group/${logicalName}
    ScalableDimension: 'elasticache:replication-group:Replicas'
    ServiceNamespace: elasticache
    RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"

ScalingPolicy:
  Type: "AWS::ApplicationAutoScaling::ScalingPolicy"
  Properties:
    ScalingTargetId: !Ref ScalingTarget
    ServiceNamespace: elasticache
    PolicyName: testpolicy
    PolicyType: TargetTrackingScaling
    ScalableDimension: 'elasticache:replication-group:Replicas'
    TargetTrackingScalingPolicyConfiguration:
      PredefinedMetricSpecification:
        PredefinedMetricType: ElastiCacheReplicaEngineCPUUtilization
      TargetValue: 40
```

Dimensionamento programmato

Il dimensionamento basato su una pianificazione consente di dimensionare le applicazioni in relazione alle variazioni di domanda prevedibili. Per utilizzare il dimensionamento pianificato, è possibile creare operazioni pianificate che indicano al ElastiCache for Redis di eseguire attività di dimensionamento a orari specifici. Al momento della creazione di un'operazione pianificata, è possibile specificare un cluster ElastiCache for Redis esistente, quando l'attività di dimensionamento dovrà verificarsi, la capacità minima e la capacità massima. È possibile creare operazioni pianificate sia una tantum che ricorrenti.

È possibile creare operazioni pianificate solo per i cluster ElastiCache for Redis già esistenti. Non è possibile creare operazioni pianificate contemporaneamente alla creazione di un cluster.

Per ulteriori informazioni sulla terminologia per la creazione, la gestione ed l'eliminazione di azioni pianificate, consulta [comandi comunemente utilizzati per creazione, la gestione ed eliminazione](#)

Per creare un'operazione pianificata una tantum

Simile alla dimensione di Partizione. Per informazioni, consultare [Dimensionamento programmato](#) .

Per eliminare un'operazione pianificata

Simile alla dimensione Partizione. Per informazioni, consultare [Dimensionamento programmato](#) .

Per gestire il dimensionamento pianificato tramite AWS CLI

Utilizza le seguenti API di scalabilità automatica delle applicazioni:

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Utilizza AWS CloudFormation Per creare criteri di Auto Scaling

Questo frammento mostra come creare un'operazione pianificata e applicarla a una risorsa [AWS::ElastiCache::ReplicationGroup](#) utilizzando la risorsa [AWS::ApplicationAutoScaling::ScalableTarget](#). Utilizza le funzioni intrinseche [Fn::Join](#) e [Ref](#) per costruire la proprietà ResourceId con il nome logico della risorsa [AWS::ElastiCache::ReplicationGroup](#) specificata nello stesso modello.

```
ScalingTarget:
  Type: 'AWS::ApplicationAutoScaling::ScalableTarget'
  Properties:
    MaxCapacity: 0
    MinCapacity: 0
    ResourceId: !Sub replication-group/${logicalName}
    ScalableDimension: 'elasticache:replication-group:Replicas'
    ServiceNamespace: elasticache
    RoleARN: !Sub "arn:aws:iam::${AWS::AccountId}:role/aws-
service-role/elasticache.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG"
    ScheduledActions:
      - EndTime: '2020-12-31T12:00:00.000Z'
        ScalableTargetAction:
          MaxCapacity: '5'
          MinCapacity: '2'
        ScheduledActionName: First
```

```
Schedule: 'cron(0 18 * * ? *)'
```

Modifica della modalità cluster

Redis è un database in memoria distribuito che supporta la partizione e la replica. I cluster ElastiCache per Redis sono l'implementazione distribuita di Redis che consente il partizionamento dei dati su più nodi Redis. Un cluster ElastiCache per Redis dispone di due modalità operative: Modalità cluster abilitata (CME) e Modalità cluster disabilitata (CMD). In CME, Redis funziona come un database distribuito con più partizioni e nodi, mentre in CMD, Redis funziona come un nodo singolo.

Prima di eseguire la migrazione da CMD a CME, devono essere soddisfatte le condizioni seguenti:

Important

La configurazione della modalità cluster può essere modificata solo dalla modalità cluster disabilitata alla modalità cluster abilitata. Non è possibile ripristinare questa configurazione.

- Il cluster può contenere chiavi solo nel database 0.
- Le applicazioni devono utilizzare un client Redis in grado di utilizzare il protocollo Cluster e utilizzare un endpoint di configurazione.
- Il failover automatico deve essere abilitato sul cluster con almeno 1 replica.
- La versione minima del motore Redis richiesta per la migrazione è 7.0.

Per eseguire la migrazione da CMD a CME, la configurazione della modalità cluster può essere modificata solo dalla modalità cluster disabilitata alla modalità cluster abilitata. Si tratta di una procedura in due fasi che garantisce la disponibilità del cluster durante il processo di migrazione.

Note

È necessario fornire un gruppo di parametri con una configurazione abilitata per i cluster, ovvero, il parametro abilitato per i cluster è impostato come yes. Se si utilizza un gruppo di parametri predefinito, ElastiCache per Redis selezionerà automaticamente il gruppo di parametri predefinito corrispondente con una configurazione abilitata per i cluster. Il valore del parametro abilitato per i cluster è impostato su no per un cluster CMD. Quando il cluster

passa alla modalità compatibile, il valore del parametro abilitato per i cluster viene aggiornato a `yes` come parte dell'azione di modifica.

Per ulteriori informazioni, consulta [Configurazione dei parametri di motore con i gruppi di parametri](#)

1. Preparazione: crea un cluster CME di prova e assicurati che lo stack sia pronto per utilizzarlo. ElastiCache per Redis non dispone di un metodo per verificare la conformità. Per ulteriori informazioni, consulta [Creazione di un cluster](#).
2. Modifica della configurazione del cluster CMD esistente alla modalità cluster: in questa modalità, verrà distribuita una partizione singola ed ElastiCache per Redis funzionerà come un nodo singolo ma anche come un cluster di partizione singolo. Modalità compatibile significa che l'applicazione client può utilizzare entrambi i protocolli per comunicare con il cluster. In questa modalità, le applicazioni devono essere riconfigurate per iniziare a utilizzare il protocollo cluster Redis e l'endpoint di configurazione. Per modificare la modalità cluster Redis in una modalità cluster compatibile, procedi nel modo seguente:

Note

In modalità compatibile, altre operazioni di modifica come il dimensionamento e la versione del motore non sono consentite per il cluster. Inoltre, i parametri (ad eccezione di `cacheParameterGroupName`) non possono essere modificati durante la definizione del parametro in modalità cluster all'interno della richiesta [ModifyReplicationGroup](#).

- a. Utilizzando la AWS Management Console, consulta [Modifica di un gruppo di replica](#) e imposta la modalità cluster su `Compatible`.
- b. Utilizzando l'API, consultare [ModifyReplicationGroup](#) e aggiornare il parametro `ClusterMode` a `compatible`.
- c. Utilizzando la AWS CLI, consultare [modify-replication-group](#) e aggiornare il parametro `cluster-mode` a `compatible`.

Dopo aver modificato la modalità cluster Redis in una modalità cluster compatibile, l'API [DescribeReplicationGroups](#) restituirà l'endpoint di configurazione del cluster ElastiCache per Redis. L'endpoint di configurazione del cluster è un endpoint singolo che può essere utilizzato

dalle applicazioni per connettersi al cluster. Per ulteriori informazioni, consulta [Individuazione degli endpoint di connessione](#).

3. Modifica della configurazione del cluster nella modalità cluster abilitata: una volta impostata la modalità cluster sulla modalità cluster compatibile, il secondo passaggio consiste nel modificare la configurazione del cluster in modalità cluster abilitata. In questa modalità, viene eseguita una partizione singola e i clienti possono ora scalare i propri cluster o modificare altre configurazioni del cluster.

Per cambiare la modalità cluster in abilitata, procedi nel modo seguente:

Prima di iniziare, assicurati che i client Redis siano stati migrati all'utilizzo del protocollo cluster e che l'endpoint di configurazione del cluster non sia in uso.

- a. Utilizzando la AWS Management Console, consulta [Modifica di un gruppo di replica](#) e imposta la modalità cluster su Abilitato.
- b. Utilizzando l'API, consultare [ModifyReplicationGroup](#) e aggiornare il parametro `ClusterMode` a `enabled`.
- c. Utilizzando la AWS CLI, consultare [modify-replication-group](#) e aggiornare il parametro `cluster-mode` a `enabled`.

Dopo aver modificato la modalità cluster in abilitata, gli endpoint verranno configurati secondo le specifiche del cluster Redis. L'API [DescribeReplicationGroups](#) restituirà il parametro della modalità cluster come `enabled` e gli endpoint del cluster che sono ora disponibili per essere utilizzati dalle applicazioni per connettersi al cluster.

Tieni presente che gli endpoint del cluster cambieranno dopo che la modalità cluster viene modificata in abilitata. Assicurati di aggiornare le applicazioni con i nuovi endpoint.

Puoi anche scegliere di tornare alla modalità cluster disabilitata (CMD) dalla modalità cluster compatibile e preservare le configurazioni originali.

Modifica della configurazione del cluster nella modalità cluster disabilitata dalla modalità cluster compatibile

1. Utilizzando la AWS Management Console, consulta [Modifica di un gruppo di replica](#) e imposta la modalità cluster su Disabilitato.

2. Utilizzando l'API, consultare [ModifyReplicationGroup](#) e aggiornare il parametro `ClusterMode` a `disabled`.
3. Utilizzando la AWS CLI, consultare [modify-replication-group](#) e aggiornare il parametro `cluster-mode` a `disabled`.

Dopo aver modificato la modalità cluster in disabilitata, l'API [DescribeReplicationGroups](#) restituirà il parametro della modalità cluster come `disabled`.

Replica tra regioni AWS mediante Global Datastore

Note

Global Datastore è attualmente disponibile solo per i cluster progettati autonomamente.

Utilizzando la caratteristica Global Datastore for Redis, è possibile utilizzare la replica completamente gestita, rapida, affidabile e sicura in tutte le regioni AWS. Utilizzando questa caratteristica è possibile creare cluster di replica di lettura tra regioni per ElastiCache for Redis per abilitare letture a bassa latenza e ripristino di emergenza nelle regioni AWS.

Nelle sezioni seguenti è possibile trovare una descrizione di come utilizzare i Global Datastore.

Argomenti

- [Panoramica](#)
- [Prerequisiti e limitazioni](#)
- [Utilizzo di Global Datastore \(Console\)](#)
- [Utilizzo di Global Datastore \(CLI\)](#)

Panoramica

Ogni Global Datastore è una raccolta di uno o più cluster che si replicano tra loro.

Un Global Datastore è costituito dai seguenti elementi:

- Cluster primario (attivo)— Un cluster primario accetta scritture replicate in tutti i cluster all'interno del datastore globale. Un cluster primario accetta anche le richieste di lettura.

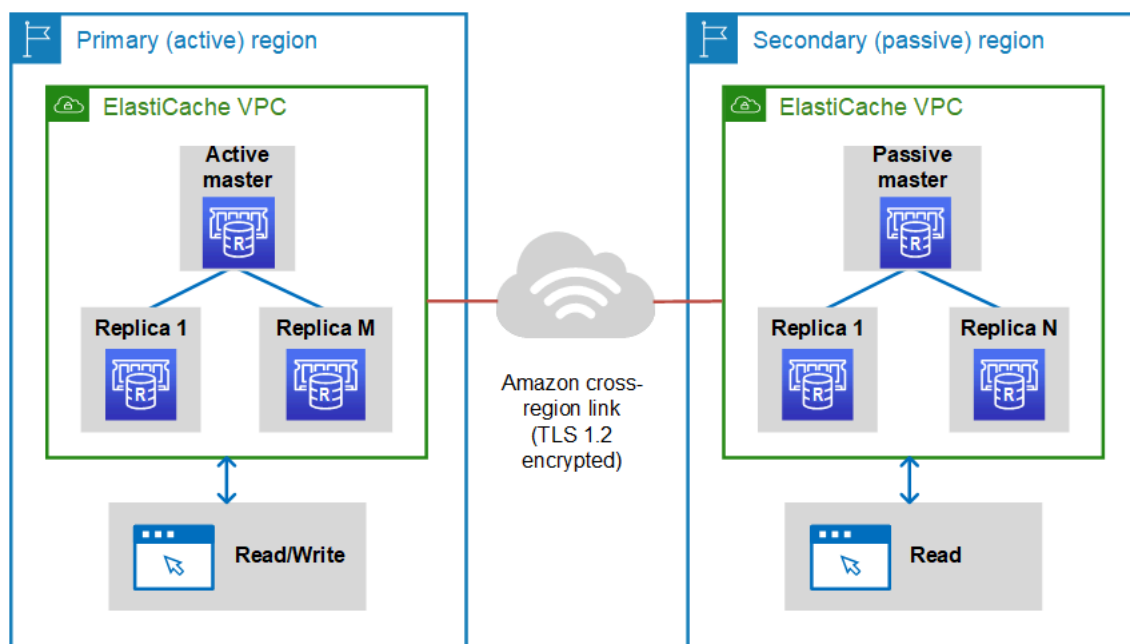
- **Cluster secondario (passivo)**— Un cluster secondario accetta solo richieste di lettura e replica gli aggiornamenti dei dati da un cluster primario. Un cluster secondario deve trovarsi in una regione AWS diversa rispetto al cluster primario.

Quando si crea un Global Datastore in ElastiCache, ElastiCache replica automaticamente i dati del cluster primario nel cluster secondario. Scegliere la regione AWS in cui i dati Redis devono essere replicati e quindi creare un cluster secondario in tale regione AWS. ElastiCache quindi configura e gestisce la replica asincrona automatica dei dati tra i due cluster.

L'utilizzo di un Global Datastore for Redis offre i seguenti vantaggi:

- **Prestazioni geolocali**— Impostando cluster di replica remoti in ulteriori regioni AWS e sincronizzando i dati tra di loro, è possibile ridurre la latenza dell'accesso ai dati in tale Regione AWS. Un Global Datastore può contribuire ad aumentare la velocità di risposta dell'applicazione fornendo letture geolocali a bassa latenza nelle regioni AWS.
- **Ripristino di emergenza**: se il cluster primario in un datastore globale subisce una degradazione, è possibile promuovere un cluster secondario come nuovo cluster primario. È possibile farlo collegandosi a una qualsiasi regione AWS che contenga un cluster secondario.

Il diagramma seguente mostra come possono funzionare i Global Datastore.



Prerequisiti e limitazioni

Prima di iniziare a utilizzare i Global Datastore, tenere presente quanto segue:

- I datastore globali sono supportati nelle seguenti AWS regioni: Asia Pacifico (Seul, Tokyo, Singapore, Sydney, Mumbai e Osaka), Europa (Francoforte, Parigi, Londra, Irlanda e Stoccolma), Stati Uniti orientali (Virginia settentrionale e Ohio), Stati Uniti occidentali (California settentrionale e Oregon), Sud America (San Paolo), (Stati Uniti occidentali e Stati Uniti orientali), Canada (regione centrale), Cina AWS GovCloud (Pechino e Ningxx) lia)
- Tutti i cluster, primari e secondari, nel datastore globale, devono avere lo stesso numero di nodi primari, tipo di nodo, versione del motore e numero di frammenti (in caso di modalità cluster abilitata). Ogni cluster nel Global Datastore può avere un numero diverso di repliche di lettura per gestire il traffico di lettura locale al cluster.

La replica deve essere abilitata se si prevede di utilizzare un cluster a nodo singolo esistente.

- I datastore globali non sono supportati su istanze precedenti a m5 o r5.
- È possibile configurare la replica per un cluster primario da una AWS regione a un cluster secondario in un massimo di altre due regioni. AWS

Note

Le eccezioni sono le regioni Cina (Pechino) Regione e Cina (Ningxia), dove la replica può avvenire solo tra le due regioni.

- È possibile utilizzare i Global Datastore solo nei cluster VPC. Per ulteriori informazioni, consulta [Modelli di accesso a una cache ElastiCache in un Amazon VPC](#). I Global Datastore non sono supportati quando si utilizza EC2-Classic. Per ulteriori informazioni, consulta [EC2-Classic](#) nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Note

Al momento, non è possibile utilizzare gli archivi dati globali in [Utilizzo di local zones con ElastiCache](#).

- ElastiCache non supporta il failover automatico da una regione all'altra AWS. Se necessario, è possibile promuovere manualmente un cluster secondario. Per vedere un esempio, consulta [Promozione del cluster secondario a primario](#).

- Per eseguire il bootstrap dai dati esistenti, utilizzare un cluster esistente come primario per creare un Global Datastore. Non supportiamo l'aggiunta di un cluster esistente come cluster secondario. Il processo di aggiunta del cluster come secondario elimina i dati e quindi può causare la perdita di dati.
- Gli aggiornamenti dei parametri vengono applicati a tutti i cluster quando si modifica un gruppo di parametri locale di un cluster appartenente a un Global Datastore.
- È possibile dimensionare i cluster regionali sia verticalmente (dimensionamento verticale) che orizzontalmente (dimensionamento orizzontale). È possibile dimensionare i cluster modificando il Global Datastore. Tutti i cluster regionali nel Global Datastore vengono quindi dimensionati senza interruzioni. Per ulteriori informazioni, consulta [Scalabilità per Redis ElastiCache](#).
- I datastore globali supportano [crittografia dei dati inattivi](#), [crittografia dei dati in transito](#) e [Redis AUTH](#).
- I datastore globali non supportano il protocollo Internet versione 6 (IPv6).
- I datastore globali supportano le chiavi. AWS KMS Per ulteriori informazioni, consulta [AWS concetti chiave di servizi gestiti](#) nella AWS Key Management Service Guida per gli sviluppatori.

Note

I Global Datastore supportano la [messaggistica PUB/SUB](#) con le seguenti disposizioni:

- Per la modalità cluster disabilitata, PUB/SUB è completamente supportata. Gli eventi pubblicati sul cluster primario della AWS regione primaria vengono propagati alle regioni secondarie. AWS
- Per la modalità cluster abilitata, si applica quanto segue:
 - Per gli eventi pubblicati che non si trovano in un keyspace, solo gli abbonati della stessa AWS regione ricevono gli eventi.
 - Per gli eventi keyspace pubblicati, gli abbonati di tutte le AWS regioni ricevono gli eventi.

Utilizzo di Global Datastore (Console)

Per creare un Global Datastore utilizzando la console, attenersi a questa procedura in due fasi:

1. Creare un cluster primario, utilizzando un cluster esistente o creandone uno nuovo. Il motore deve essere Redis 5.0.6 o versioni successive.

2. Aggiungi fino a due cluster secondari in AWS regioni diverse, sempre utilizzando il motore Redis 5.0.6 o successivo.

Le seguenti procedure ti guidano su come creare un datastore globale per Redis ed eseguire altre operazioni utilizzando la console for Redis. ElastiCache

Argomenti

- [Creazione di un Global Datastore utilizzando un cluster esistente](#)
- [Creazione di un nuovo Global Datastore utilizzando un nuovo cluster primario](#)
- [Visualizzazione dei dettagli del Datastore globale](#)
- [Aggiunta di una regione a un Global Datastore](#)
- [Modifica di un Global Datastore](#)
- [Promozione del cluster secondario a primario](#)
- [Rimozione di una regione da un Global Datastore](#)
- [Eliminazione di un Global Datastore](#)

Creazione di un Global Datastore utilizzando un cluster esistente

In questo scenario, è possibile utilizzare un cluster esistente da utilizzare come cluster primario del nuovo Global Datastore. È quindi possibile creare un cluster secondario di sola lettura in una regione AWS separata. Questo cluster secondario riceve aggiornamenti automatici e asincroni dal cluster primario.

Important

Il cluster esistente deve utilizzare il motore Redis 5.0.6 o versioni successive.

Per creare un Global Datastore utilizzando un cluster esistente

1. [Accedi AWS Management Console e apri la console all'indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/). ElastiCache
2. Nel pannello di navigazione, scegliere Redis, quindi scegliere un cluster.
3. Per Actions (Operazioni), scegliere Setup Global Datastore (Configurazione Global Datastore).

4. Nella pagina Setup Global Datastore (Configurazione Global Datastore) eseguire le operazioni seguenti:
 - Immettere un valore per Global Datastore Name suffix (Suffisso nome Global Datastore): questo suffisso viene utilizzato per generare un nome univoco per il Global Datastore. È possibile cercare il Global Datastore utilizzando il suffisso specificato qui.
 - (Opzionale) Immettere un valore Description (Descrizione).
5. In Secondary cluster details (Dettagli cluster secondario), scegliere una regione AWS diversa in cui archiviare il cluster.
6. In Redis settings (Impostazioni Redis), immettere un valore per Name (Nome) e, facoltativamente, per Description (Descrizione) per il cluster.
7. Mantenere le seguenti opzioni invariate. Sono precompilate per corrispondere alla configurazione del cluster primario, non è possibile modificarle.
 - Versione del motore
 - Tipo di nodo
 - Gruppo di parametri

Note

ElastiCache genera automaticamente un nuovo gruppo di parametri dai valori del gruppo di parametri fornito e applica il nuovo gruppo di parametri al cluster. Utilizzare questo nuovo gruppo di parametri per modificare i parametri in un Global Datastore. Ogni gruppo di parametri generati automaticamente è associato a un solo cluster e, pertanto, a un solo Global Datastore.

- Numero di partizioni
- 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. Per ulteriori informazioni, consulta [Utilizzo delle chiavi AWS KMS gestite dal cliente](#).

- Crittografia dei dati in transito : Consente la crittografia dei dati in trasferimento. Per maggiori informazioni, consultare [Crittografia dei dati in transito](#). Per il motore Redis versione 6.0, se si abilita la crittografia in transito, viene richiesto di specificare una delle seguenti opzioni di Controllo degli accessi:
 - Nessun controllo degli accessi - Questa è l'impostazione di default. Questo non indica alcuna restrizione.
 - Lista di controllo accessi gruppi di utenti— Scegliere un gruppo di utenti con un set definito di utenti e autorizzazioni per le operazioni disponibili. Per ulteriori informazioni, consulta [Gestione dei gruppi di utenti con Console e CLI](#).
 - Redis AUTH Default User (Utente di default Redis AUTH) : Un meccanismo di autenticazione per il server Redis. Per ulteriori informazioni, vedere [Redis AUTH](#).
8. (Opzionale) In base alle necessità, aggiornare le restanti impostazioni del cluster secondario. Queste sono precompilate con gli stessi valori del cluster primario, ma è possibile aggiornarle per soddisfare requisiti specifici di tale cluster.
- Porta
 - Numero di repliche
 - Subnet group (Gruppo di sottoreti)
 - Zone di disponibilità preferite
 - Gruppi di sicurezza
 - Gestito dal cliente (chiave AWS KMS)
 - Token Redis AUTH
 - Abilitazione backup automatici.
 - Backup retention period (Periodo di retention dei backup)
 - Finestra di backup
 - Maintenance window (Finestra di manutenzione)
 - Argomento per la notifica SNS
9. Scegli Crea. In questo modo lo stato del Global Datastore viene impostato su Creating (Creazione in corso). Lo stato passa a Modifying (Modifica in corso) dopo che il cluster primario viene associato al Global Datastore e il cluster secondario è in stato Associating (Associazione in corso).

Dopo che il cluster primario e i cluster secondari sono stati associati al Global Datastore, lo stato cambia in Available (Disponibile). A questo punto, si dispone di un cluster primario che accetta letture e scritture e cluster secondari che accettano letture replicate dal cluster primario.

La pagina Redis viene aggiornata per indicare se un cluster fa parte di un Global Datastore, come segue:

- Global Datastore— Il nome del Datastore globale a cui appartiene il cluster.
- Ruolo Global Datastore— Il ruolo del cluster, primario o secondario.

È possibile aggiungere fino a un cluster secondario aggiuntivo in una AWS regione diversa. Per ulteriori informazioni, consulta [Aggiunta di una regione a un Global Datastore](#).

Creazione di un nuovo Global Datastore utilizzando un nuovo cluster primario

Se si sceglie di creare un Datastore globale con un nuovo cluster, utilizzare la procedura seguente.

1. Accedi AWS Management Console e apri la ElastiCache console all'[indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Nel riquadro di navigazione scegli Datastore globale e seleziona Crea datastore globale.
3. Sotto Impostazioni cluster primario, procedi come indicato di seguito:
 - a. Per Modalità cluster scegli Abilitato o Disabilitato.
 - b. Per informazioni su Global Datastore, inserisci un valore per Nome. ElastiCache utilizza il suffisso per generare un nome univoco per il datastore globale. È possibile cercare il Global Datastore utilizzando il suffisso specificato qui.
 - c. (Opzionale) Immettere un valore per Global Datastore Description (Descrizione Global Datastore).
4. Sotto Cluster regionale:
 - a. Per Regione, scegli una regione disponibile. AWS
 - b. Scegli Crea nuovo cluster regionale o Usa cluster regionale esistente
 - c. Se scegli Crea nuovo cluster regionale, sotto Informazioni sul cluster inserisci un nome e una descrizione facoltativa del cluster.
 - d. Sotto Posizione, è preferibile accettare le impostazioni predefinite per Multi-AZ e Failover automatico.

5. Sotto Impostazioni cluster

- a. Per Versione motore scegli una versione disponibile, cioè 5.0.6 o successiva.
- b. Per Porta utilizza la porta predefinita, 6379. Se esiste un motivo per utilizzare una porta diversa, immettere il numero di porta.
- c. Per Parameter group (Gruppo di parametri), scegliere un gruppo di parametri o crearne uno nuovo. I gruppi di parametri controllano i parametri di runtime del cluster. Per ulteriori informazioni sui gruppi di parametri, consulta [Parametri specifici di Redis](#) e [Creazione di un gruppo di parametri](#).

Note

Quando si seleziona un gruppo di parametri per impostare i valori di configurazione del motore, tale gruppo di parametri viene applicato a tutti i cluster nel Global Datastore. Nella pagina Parameter Groups (Gruppi di parametri) con/senza l'attributo Global (Globale) indica se un gruppo di parametri fa parte di un Global Datastore.

- d. In Node type (Tipo di nodo), scegliere la freccia in giù (▼).

Nella finestra di dialogo Change node type (Cambia tipo di nodo) scegliere un valore per Instance family (Famiglia di istanze) per il tipo di nodo desiderato. Scegliere quindi il tipo di nodo che si desidera utilizzare per il cluster, quindi scegliere Save (Salva).

Per ulteriori informazioni, consulta [Scelta delle dimensioni dei nodi](#).

Se scegli un tipo di nodo r6gd, il tiering dei dati viene abilitato automaticamente. Per ulteriori informazioni, consulta [Tiering di dati](#).

- e. Se stai creando un cluster Redis (modalità cluster disabilitata):

In Numero di repliche scegli il numero di repliche per questo cluster.

- f. Se stai creando un cluster Redis (modalità cluster abilitata):

- i. In Number of Shards (Numero di partizioni), scegliere il numero di partizioni (partizioni/gruppi di nodi) per questo cluster Redis (modalità cluster abilitata).

Per alcune versioni di Redis (modalità cluster abilitata), è possibile modificare dinamicamente il numero di partizioni nel cluster:

- Redis 3.2.10 e versioni successive – Se il cluster esegue Redis 3.2.10 o versioni successive, è possibile modificare dinamicamente il numero delle partizioni nel cluster. Per ulteriori informazioni, consulta [Ridimensionamento dei cluster in Redis \(modalità cluster attivata\)](#).
 - Altre versioni di Redis – Se il cluster esegue una versione di Redis precedente alla versione 3.2.10, esiste un altro approccio. Per modificare il numero delle partizioni nel cluster in questo caso, creare un nuovo cluster con il nuovo numero delle partizioni. Per ulteriori informazioni, consulta [Ripristino da un backup in una nuova cache](#).
- ii. In Replicas per shard (Repliche per partizione): scegliere il numero di nodi di replica di lettura per ogni partizione.

Esistono le seguenti restrizioni per Redis (modalità cluster abilitata).

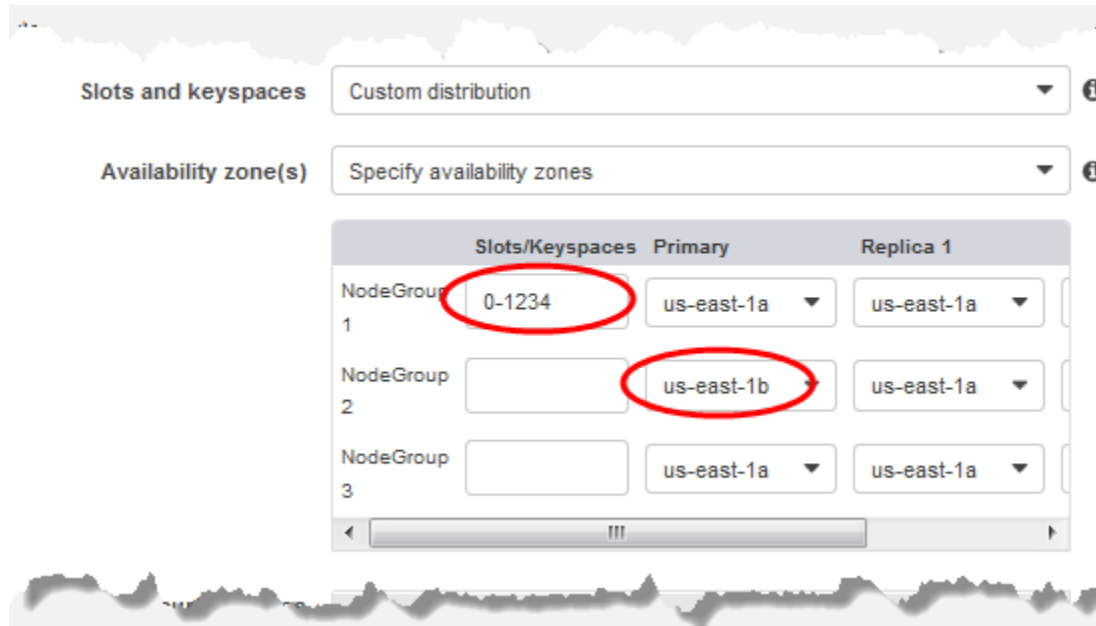
- Se hai abilitato la funzione Multi-AZ, assicurati di avere almeno una replica per ogni partizione.
 - Quando utilizzi la console per creare il cluster, il numero delle repliche è lo stesso per ogni partizione.
 - Il numero delle repliche di lettura per ogni partizione è fisso e non modificabile. Se scopri che ti servono più o meno repliche per partizione (API/CLI: gruppo di nodi), devi creare un nuovo cluster con il nuovo numero di repliche. Per ulteriori informazioni, consulta [Inizializzazione di un nuovo cluster progettato autonomamente con un backup creato esternamente](#).
6. Per le impostazioni del gruppo di sottoreti, scegli la sottorete che desideri applicare a questo cluster. ElastiCache fornisce un gruppo di sottoreti IPv4 predefinito oppure puoi scegliere di crearne uno nuovo. Per IPv6 devi creare un gruppo di sottoreti con un blocco CIDR IPv6. Se scegli dual-stack, devi selezionare un tipo di individuazione IP: IPv6 o IPv4.

Per ulteriori informazioni, consulta [Creazione di una sottorete nel VPC](#).

7. Per Posizionamento zone disponibilità, le opzioni sono due:
- Nessuna preferenza: ElastiCache sceglie la zona di disponibilità.
 - Specify Availability Zones (Specifica zone di disponibilità) – L'utente specifica la zona di disponibilità per ogni cluster.

Se hai scelto di specificare le zone di disponibilità, per ogni cluster in ogni partizione, seleziona la zona di disponibilità dall'elenco.

Per ulteriori informazioni, consulta [Scelta di regioni e zone di disponibilità](#).




Specifica dei Keyspace e delle zone di disponibilità

8. Scegli Successivo.
 9. In Impostazioni avanzate di Redis
 - Per Sicurezza:
 - i. 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 scegliendo la chiave AWS KMS gestita dal cliente e scegliendo la chiave. Per ulteriori informazioni, vedere [Utilizzo di chiavi gestite dal cliente da AWS KMS](#).
- Crittografia dei dati in transito – Consente la crittografia dei dati in trasferimento. Per maggiori informazioni, consultare [Crittografia dei dati in transito](#). Per il motore Redis versione 6.0 e successive, se si abilita la Crittografia in transito, viene richiesto di specificare una delle seguenti opzioni di Controllo degli accessi:

- Nessun controllo degli accessi - Questa è l'impostazione di default. Questo non indica alcuna restrizione sull'accesso dell'utente al cluster.
- Lista di controllo accessi – scegliere un gruppo di utenti con un set definito di utenti che possono accedere al cluster. Per ulteriori informazioni, consulta [Gestione dei gruppi di utenti con Console e CLI](#).
- Redis AUTH Default User (Utente di default Redis AUTH) : Un meccanismo di autenticazione per il server Redis. Per ulteriori informazioni, vedere [Redis AUTH](#).
- Redis AUTH - Un meccanismo di autenticazione per il server Redis. Per ulteriori informazioni, vedere [Redis AUTH](#).

 Note

Per le versioni di Redis a partire da 3.2.6 e successive, esclusa la versione 3.2.10, Redis AUTH è l'unica opzione.

- ii. 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.

10. Se si desiderano backup regolari pianificati regolarmente, scegliere Enable automatic backups (Abilita backup automatici), quindi digitare il numero di giorni in cui conservare ciascun backup automatico prima che venga automaticamente eliminato. Se non si desiderano backup automatici pianificati regolarmente, deselezionare la casella di controllo Enable automatic backups (Abilita backup automatici). In ogni caso, hai sempre la possibilità di creare backup manuali.

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

11. (Facoltativo) Specifica una finestra di manutenzione. La Maintenance window (Finestra di manutenzione) è l'intervallo di tempo, generalmente un'ora a settimana, in cui ElastiCache pianifica la manutenzione di sistema per il cluster. Puoi ElastiCache scegliere il giorno e l'ora della finestra di manutenzione (nessuna preferenza) oppure puoi scegliere tu stesso 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](#).

12. (Facoltativo) PerRegistro:

- In Formato del registro, scegliere Text o JSON.
- In Tipo di destinazione, scegli CloudWatch Logs o Kinesis Firehose.
- In Destinazione log, scegliete Crea nuovo e inserite il nome del gruppo di log CloudWatch Logs o il nome dello stream Firehose, oppure scegliete Seleziona esistente e quindi scegliete il nome del gruppo di log Logs o CloudWatch il nome dello stream Firehose,

13. Per quanto riguarda i tag, per aiutarti a gestire i tuoi cluster e altre ElastiCache risorse, puoi assegnare i tuoi metadati a ciascuna risorsa sotto forma di tag. Per ulteriori informazioni, consulta [Assegnazione di tag delle risorse ElastiCache](#).

14. Riesaminare le voci e le selezioni, quindi apportare le eventuali correzioni. Al termine, scegli Next (Successivo).


15. Dopo aver configurato il cluster nei passaggi precedenti, ora puoi configurare i dettagli dei cluster secondari.

16. In Cluster regionale, scegli la AWS regione in cui si trova il cluster.

17. Sotto Informazioni sul cluster inserisci un nome e una descrizione facoltativa del cluster.

18. Le opzioni seguenti sono precompilate in modo che corrispondano alla configurazione del cluster primario e non possono essere modificate:


- Ubicazione
- Versione del motore
- Tipo di istanza
- Tipo di nodo
- Numero dle partizioni
- Gruppo di parametri

 Note

ElastiCache genera automaticamente un nuovo gruppo di parametri dai valori del gruppo di parametri fornito e applica il nuovo gruppo di parametri al cluster. Utilizzare questo nuovo gruppo di parametri per modificare i parametri in un Global Datastore.


Ogni gruppo di parametri generati automaticamente è associato a un solo cluster e, pertanto, a un solo Global Datastore.

- 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. Per ulteriori informazioni, consulta [Utilizzo delle chiavi AWS KMS gestite dal cliente](#).

- Crittografia dei dati in transito : Consente la crittografia dei dati in trasferimento. Per maggiori informazioni, consultare [Crittografia dei dati in transito](#). Per il motore Redis versione 6.4 e superiore, se abiliti la crittografia in transito, ti viene chiesto di specificare una delle seguenti opzioni Controllo accessi:
 - Nessun controllo degli accessi - Questa è l'impostazione di default. Questo non indica alcuna restrizione sull'accesso dell'utente al cluster.
 - Lista di controllo accessi - Scegliere un gruppo di utenti con un set definito di utenti che possono accedere al cluster. Per ulteriori informazioni, consulta [Gestione dei gruppi di utenti con Console e CLI](#).
 - Redis AUTH Default User (Utente di default Redis AUTH) : Un meccanismo di autenticazione per il server Redis. Per ulteriori informazioni, vedere [Redis AUTH](#).

 Note

Per le versioni Redis tra 4.0.2, quando la crittografia in transito è stata supportata per la prima volta, e 6.0.4, Redis AUTH è l'unica opzione.

Le restanti impostazioni del cluster secondario vengono precompilate con gli stessi valori del cluster primario, ma è possibile aggiornare quanto segue per soddisfare requisiti specifici per tale cluster:

- Porta
- Numero di repliche

- Subnet group (Gruppo di sottoreti)
 - Zone di disponibilità preferite
 - Gruppi di sicurezza
 - Gestito dal cliente (chiave AWS KMS)
 - Token Redis AUTH
 - Abilitazione backup automatici.
 - Backup retention period (Periodo di retention dei backup)
 - Finestra di backup
 - Maintenance window (Finestra di manutenzione)
 - Argomento per la notifica SNS
19. Scegli Crea. Ciò imposta lo stato del Global Datastore su Creating (Creazione in corso). Dopo che il cluster primario e i cluster secondari sono stati associati al Global Datastore, lo stato cambia in Available (Disponibile). Si dispone di un cluster primario che accetta letture e scritture e di un cluster secondario che accetta letture replicate dal cluster primario.

La pagina Redis viene inoltre aggiornata per indicare se un cluster fa parte di un Global Datastore, inclusi i seguenti elementi:

- Global Datastore— Il nome del Datastore globale a cui appartiene il cluster.
- Ruolo Global Datastore— Il ruolo del cluster, primario o secondario.

È possibile aggiungere fino a un cluster secondario aggiuntivo in una AWS regione diversa. Per ulteriori informazioni, consulta [Aggiunta di una regione a un Global Datastore](#).

Visualizzazione dei dettagli del Datastore globale

È possibile visualizzare i dettagli dei Global Datastore esistenti e anche modificarli nella pagina Global Datastore.

Per visualizzare i dettagli del Global Datastore

1. Accedi AWS Management Console e apri la ElastiCache console all'[indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Nel pannello di navigazione, scegli Global Datastore, quindi scegli un Global Datastore disponibile.

È quindi possibile esaminare le seguenti proprietà del Global Datastore:

- Global Datastore Name (Nome Global Datastore: il nome del Global Datastore)
- Description (Descrizione): una descrizione del Global Datastore
- Status (Stato): le opzioni includono:
 - Creazione
 - Modifying (Modifica in corso)
 - Disponibilità
 - Eliminazione in corso
 - Primary-Only (Solo primario): questo stato indica che il Global Datastore contiene solo un cluster primario. Tutti i cluster secondari vengono eliminati o creati non correttamente.
- Cluster Mode (Modalità cluster): abilitata o disabilitata
- Redis Engine Version (Versione del motore Redis): la versione del motore Redis che esegue il Global Datastore
- Instance Node Type (Tipo di nodo di istanza): il tipo di nodo utilizzato per il Global Datastore
- Encryption at-rest (Crittografia dei dati inattivi): abilitata o disabilitata
- Encryption in-transit (Crittografia dei dati in transito): abilitata o disabilitata
- Redis AUTH: abilitata o disabilitata

È possibile apportare le seguenti modifiche al Global Datastore:

- [Aggiunta di una regione a un Global Datastore](#)
- [Rimozione di una regione da un Global Datastore](#)
- [Promozione del cluster secondario a primario](#)
- [Modifica di un Global Datastore](#)

La pagina Global Datastore elenca anche i singoli cluster che formano il Global Datastore e le seguenti proprietà per ciascuno:

- Regione: la AWS regione in cui è archiviato il cluster
- Role (Ruolo): primario o secondario
- Cluster Name (Nome cluster): il nome del cluster
- Status (Stato): le opzioni includono:

- **Associating (Associazione in corso):** il cluster è in procinto di essere associato al Global Datastore
- **Associated (Associato):** il cluster è stato associato al Global Datastore
- **Disassociating (Annullamento associazione):** il processo di rimozione di un cluster secondario dal Global Datastore utilizzando il nome del Global Datastore. Dopodiché, il cluster secondario non riceve più aggiornamenti dal cluster primario ma rimane un cluster autonomo in quella AWS regione.
- **Disassociated (Disassociato):** il cluster secondario è stato rimosso dal Global Datastore ed è ora un cluster autonomo nella propria regione AWS .
- **Global Datastore Replica lag:** mostra un valore per AWS regione secondaria nel datastore globale. Questo è il ritardo tra il nodo primario della regione secondaria e il nodo primario della regione primaria. Per Redis abilitato in modalità cluster, il ritardo indica il ritardo massimo tra le partizioni.


Aggiunta di una regione a un Global Datastore

È possibile aggiungere fino a una AWS regione aggiuntiva a un datastore globale esistente. In questo scenario, si sta creando un cluster di sola lettura in una AWS regione separata che riceve aggiornamenti automatici e asincroni dal cluster primario.

Per aggiungere una regione a un AWS datastore globale


1. [Accedi AWS Management Console e apri la ElastiCache console all'indirizzo https://console.aws.amazon.com/elasticache/.](https://console.aws.amazon.com/elasticache/)
2. Nel pannello di navigazione scegliere Global Datastore quindi per Global Datastore Name (Nome datastore globale, scegliere un datastore globale.
3. Scegli Aggiungi regione e scegli la AWS regione in cui deve risiedere il cluster secondario.
4. In Redis settings (Impostazioni Redis), immettere un valore per Name (Nome) e, facoltativamente, per Description (Descrizione) per il cluster.
5. Mantenere le seguenti opzioni invariate. Sono precompilate per corrispondere alla configurazione del cluster primario, non è possibile modificarle.
 - Versione del motore
 - Tipo di istanza
 - Tipo di nodo
 - Numero dle partizioni

- Gruppo di parametri

 Note

ElastiCache genera automaticamente un nuovo gruppo di parametri dai valori del gruppo di parametri fornito e applica il nuovo gruppo di parametri al cluster. Utilizzare questo nuovo gruppo di parametri per modificare i parametri in un Global Datastore. Ogni gruppo di parametri generati automaticamente è associato a un solo cluster e, pertanto, a un solo Global Datastore.

- Crittografia a riposo

 Note

È possibile fornire una chiave di crittografia diversa scegliendo la chiave AWS KMS gestita dal cliente e scegliendo la chiave.

- Crittografia in transito

- Redis AUTH

6. (Opzionale) Aggiornare le restanti impostazioni del cluster secondario. Queste sono precompilate con gli stessi valori del cluster primario, ma è possibile aggiornarle per soddisfare requisiti specifici di tale cluster:

- Porta
- Numero di repliche
- Subnet group (Gruppo di sottoreti)
- Zone di disponibilità preferite
- Gruppi di sicurezza
- Chiave AWS KMS gestita dal cliente)
- Token Redis AUTH
- Abilitazione backup automatici.
- Backup retention period (Periodo di retention dei backup)
- Finestra di backup
- Maintenance window (Finestra di manutenzione)

- Argomento per la notifica SNS

7. Scegliere Aggiungi.

Modifica di un Global Datastore

È possibile modificare le proprietà dei cluster regionali. Su un Global Datastore può essere in corso una sola operazione di modifica, ad eccezione della promozione di un cluster secondario a primario. Per ulteriori informazioni, consulta [Promozione del cluster secondario a primario](#).

Per modificare un Global Datastore

1. Accedi AWS Management Console e apri la ElastiCache console all'[indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Nel pannello di navigazione scegliere Global Datastore quindi per Nome Global Datastore, scegliere un Global Datastore.
3. Scegliere Modifica e scegliere tra le seguenti opzioni:
 - Descrizione: una descrizione del Global Datastore
 - Modifica della versione del motore— È disponibile solo il motore Redis versione 5.0.6 o successiva.
 - Modifica tipo di nodo: Dimensionamento dei cluster regionali sia verticalmente (dimensionamento verso l'alto e verso il basso) che orizzontalmente (dimensionamento orizzontale). Le opzioni includono le famiglie di nodi R5 e M5. Per ulteriori informazioni sui tipi di nodi, consulta [Tipi di nodi supportati](#).
 - Modifica del failover automatico— Abilita o disabilita il failover automatico. Quando si abilita il failover e i nodi primari nei cluster regionali si chiudono in modo imprevisto, ElastiCache viene eseguito il failover su una delle repliche regionali. Per maggiori informazioni consulta [Failover automatico](#).

Per i cluster Redis con la modalità cluster abilitata:

- Aggiungile partizioni: Immetti il numero dle partizioni da aggiungere e, facoltativamente, specifica una o più zone di disponibilità.
- Elimina frammenti: scegli i frammenti da eliminare in ogni regione. AWS
- Ribilanciamento dle partizioni Ribilancia la distribuzione degli slot per garantire una distribuzione uniforme tra le partizioni esistenti nel cluster.

Per modificare i parametri di un datastore globale, modifica il gruppo di parametri di qualsiasi cluster membro del datastore globale. ElastiCache applica automaticamente questa modifica a tutti i cluster all'interno di quel datastore globale. Per modificare il gruppo di parametri di quel cluster, utilizza la console Redis o l'operazione API. [ModifyCacheCluster](#) Per ulteriori informazioni, consulta [Modifica di un gruppo di parametri](#). Quando si modifica il gruppo di parametri di qualsiasi cluster contenuto in un Global Datastore, ciò si applica a tutti i cluster all'interno del Global Datastore.

Per reimpostare un intero gruppo di parametri o parametri specifici, utilizza l'operazione [ResetCacheParameterGroupAPI](#).

Promozione del cluster secondario a primario

Se il cluster o la AWS regione primari non sono disponibili o presentano problemi di prestazioni, è possibile promuovere un cluster secondario a primario. La promozione è consentita in qualsiasi momento, anche se sono in corso altre modifiche. È inoltre possibile effettuare più promozioni in parallelo e al termine il Global Datastore si risolve in un cluster primario. Se promuovi più cluster secondari contemporaneamente, ElastiCache for Redis non garantisce quale di essi alla fine diventi primario.

Per promuovere un cluster secondario a primario

1. [Accedi AWS Management Console e apri la console all'indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/). [ElastiCache](#)
2. Nel pannello di navigazione scegliere Global Datastore in Redis.
3. Scegliere il nome del Global Datastore per visualizzarne i dettagli
4. Scegliere il cluster Secondario
5. Scegliere Promuovi a primario.

Viene quindi richiesto di confermare la decisione con il seguente avviso: Promoting a region to primary will make the cluster in this region as read/writable. Are you sure you want to promote the *secondary* cluster to primary?

The current primary cluster in *primary region* will become secondary and will stop accepting writes after this operation completes. Please ensure you update your application stack to direct traffic to the new primary region.

6. Scegliere Conferma se si desidera continuare con la promozione o Cancel (Annulla) in caso contrario.

Se si sceglie di confermare, il Global Datastore passa a uno stato Modifying (Modifica in corso) e non sarà disponibile fino al completamento della promozione.

Rimozione di una regione da un Global Datastore

È possibile rimuovere una AWS regione da un datastore globale utilizzando la procedura seguente.

Per rimuovere una AWS regione da un datastore globale

1. [Accedi AWS Management Console e apri la ElastiCache console all'indirizzo https://console.aws.amazon.com/elasticache/.](https://console.aws.amazon.com/elasticache/)
2. Nel pannello di navigazione scegliere Global Datastore in Redis.
3. Scegliere un datastore globale
4. Scegliere Region (Regione) per scegliere la regione che si desidera rimuovere.
5. Scegliere Remove region (Rimuovi regione).

Note

Questa opzione è disponibile solo per i cluster secondari.

Verrà quindi chiesto di confermare la decisione con il seguente avviso: Removing the region will remove your only available cross region replica for the primary cluster. Your primary cluster will no longer be set up for disaster recovery and improved read latency in remote region. Are you sure you want to remove the selected region from the global datastore?

6. Scegliere Conferma se si desidera continuare con la promozione o Cancel (Annulla) in caso contrario.

Se scegli conferma, la AWS regione viene rimossa e il cluster secondario non riceve più gli aggiornamenti di replica.

Eliminazione di un Global Datastore

Per eliminare un Global Datastore, rimuovere innanzitutto tutti i cluster secondari. Per ulteriori informazioni, consulta [Rimozione di una regione da un Global Datastore](#). In questo modo il Global Datastore globale è in stato primary-only (solo primario).

Per eliminare un Global Datastore

1. Accedi AWS Management Console e apri la ElastiCache console all'[indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Nel pannello di navigazione scegliere Global Datastore in Redis.
3. In Global Datastore Name (Nome Global Datastore) scegliere il Global Datastore che si desidera eliminare e quindi scegliere Delete (Elimina).

Verrà quindi chiesto di confermare la decisione con il seguente avviso: Are you sure you want to delete this Global Datastore?

4. Scegli Elimina.

Il Global Datastore passa allo stato Deleting (Eliminazione in corso).

Utilizzo di Global Datastore (CLI)

È possibile utilizzare l'AWS Command Line Interface (AWS CLI) per controllare più servizi AWS dalla riga di comando e automatizzarli tramite script. È possibile utilizzare la AWS (CLI) per operazioni ad hoc (una tantum).

Download e configurazione dell' AWS CLI

L'AWS CLI funziona in Windows, macOS o Linux. Utilizzare la procedura seguente per scaricarla e configurarla.

Per scaricare, installare e configurare la CLI

1. Scaricare la CLI AWS dalla pagina Web [interfaccia a riga di comando di AWS](#).
2. Seguire le istruzioni per l'installazione della CLI AWS e la configurazione della CLI AWS nella AWS Command Line Interfaceguida per l'utente.

Utilizzo della CLI AWS con i datastore globali

Utilizzare le seguenti operazioni della CLI per utilizzare i Global Datastore:

- [create-global-replication-group](#)

```
aws elasticache create-global-replication-group \
  --global-replication-group-id-suffix my global datastore \
  --primary-replication-group-id sample-repl-group \
  --global-replication-group-description an optional description of the global
  datastore
```

Amazon ElastiCache applica automaticamente un prefisso all'ID datastore globale quando viene creato. Ciascuna AWS regione ha il suo prefisso. Ad esempio, un ID datastore globale creato nella regione Stati Uniti occidentali (California settentrionale) inizia con "virxk" assieme al nome del suffisso fornito. Il suffisso, combinato al prefisso autogenerato, garantisce l'unicità del nome del datastore globale in più Regioni.

Nella tabella seguente sono elencate tutte le regioni AWS e il relativo prefisso ID datastore globale.

Nome regione/Regione	Prefix
US East (Ohio) Region us-east-2	fpkhr
US East (N. Virginia) Region us-east-1	ldgnf
US West (N. California) Region us-west-1	virxk
US West (Oregon) Region us-west-2	sgau
Regione Canada (Centrale)	bxodz

Nome regione/Regione	Prefix
ca-central-1	
Asia Pacific (Mumbai) Region ap-south-1	erpgt
Asia Pacific (Tokyo) Region ap-northeast-1	quwsw
Regione Asia Pacifico (Seoul) ap-northeast-2	lfqnh
Regione Asia Pacifico (Osaka-Locale) ap-northeast-3	nlapn
Asia Pacific (Singapore) Region ap-southeast-1	v1qxn
Asia Pacific (Sydney) Region ap-southeast-2	vbgxd
Europe (Frankfurt) Region eu-central-1	iudkw
Europe (Ireland) Region eu-west-1	gxeiz
Europe (London) Region eu-west-2	okuqm

Nome regione/Regione	Prefix
Regione UE (Parigi) eu-west-3	fgjhi
South America (São Paulo) Region sa-east-1	juxlw
Regione Cina (Pechino) cn-north-1	emvgo
Regione Cina (Ningxia) cn-northwest-1	ckbem
Regione Asia Pacifico (Hong Kong) ap-east-1	knjmp
AWS GovCloud (Stati Uniti occidentali) us-gov-west-1	sgwui

- [create-replication-group \(crea gruppo di replica\)](#): Utilizzare questa operazione per creare cluster secondari per un datastore globale fornendo il nome del datastore globale al parametro `--global-replication-group-id`.

```
aws elasticache create-replication-group \
  --replication-group-id secondary replication group name \
  --replication-group-description "Replication group description" \
  --global-replication-group-id global datastore name
```

Quando si chiama questa operazione e si passa in un valore `--global-replication-group-id`, ElastiCache for Redis dedurrà i valori dal gruppo di replica principale del gruppo di replica globale per i parametri seguenti. Non passare valori per questi parametri:

"PrimaryClusterId",

"AutomaticFailoverEnabled",
"NumNodeGroups",
"CacheParameterGroupName",
"CacheNodeType",
"Engine",
"EngineVersion",
"CacheSecurityGroupNames",
"EnableTransitEncryption",
"AtRestEncryptionEnabled",
"SnapshotArns",
"SnapshotName"

- [describe-global-replication-groups](#)

```
aws elasticache describe-global-replication-groups \  
  --global-replication-group-id my global datastore \  
  --show-member-info an optional parameter that returns a list of the primary and  
  secondary clusters that make up the global datastore
```

- [modify-global-replication-group](#)

```
aws elasticache modify-global-replication-group \  
  --global-replication-group-id my global datastore \  
  --automatic-failover-enabled \  
  --cache-node-type node type \  
  --cache-parameter-group-name parameter group name \  
  --engine-version engine version \  
  --apply-immediately \  
  --global-replication-group-description description
```

- [delete-global-replication-group](#)

```
aws elasticache delete-global-replication-group \  
  --global-replication-group-id my global datastore \  
  --retain-primary-replication-group defaults to true
```

- [disassociate-global-replication-group](#)

```
aws elasticache disassociate-global-replication-group \  
  --global-replication-group-id my global datastore \  
  --replication-group-id my secondary cluster \  
  --replication-group-region the AWS Region in which the secondary cluster resides
```

- [failover-global-replication-group](#)

```
aws elasticache failover-replication-group \  
  --global-replication-group-id my global datastore \  
  --primary-region The AWS Region of the primary cluster \  
  --primary-replication-group-id The name of the global datastore, including the suffix.
```

- [increase-node-groups-in-global-replication-group](#)

```
aws elasticache increase-node-groups-in-global-replication-group \  
  --apply-immediately yes \  
  --global-replication-group-id global-replication-group-name \  
  --node-group-count 3
```

- [decrease-node-groups-in-global-replication-group](#)

```
aws elasticache decrease-node-groups-in-global-replication-group \  
  --apply-immediately yes \  
  --global-replication-group-id global-replication-group-name \  
  --node-group-count 3
```

- [rebalance-shards-in-global-replication-group](#)

```
aws elasticache rebalance-shards-in-global-replication-group \  
  --apply-immediately yes \  
  --global-replication-group-id global-replication-group-name
```

Utilizzare la Guida per un elenco di tutti i comandi ElastiCache for Redis disponibili.

```
aws elasticache help
```

È anche possibile utilizzare la guida per la descrizione di un comando specifico e per ulteriori informazioni relative al suo utilizzo:

```
aws elasticache create-global-replication-group help
```

Alta disponibilità utilizzando gruppi di replica

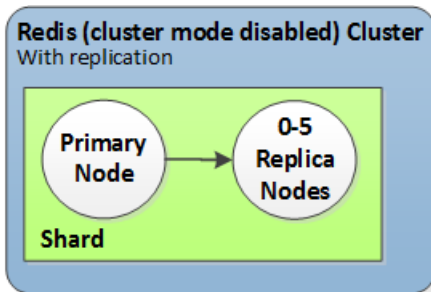
I cluster Amazon ElastiCache Redis a nodo singolo sono entità in memoria con servizi di protezione dei dati (AOF) limitati. Se il cluster non viene eseguito per qualsiasi motivo, tutti i dati del cluster vengono persi. Tuttavia, se stai eseguendo il motore Redis, puoi raggruppare da 2 a 6 nodi in un cluster con repliche in cui solo da 1 a 5 nodi di sola lettura contengono dati di replica del singolo nodo primario lettura/scrittura del gruppo. In questo scenario, se un nodo non viene eseguito per qualsiasi motivo, non tutti i dati vengono persi poiché vengono replicati in uno o più nodi. A causa della latenza di replica, alcuni dati potrebbero essere persi se è il nodo di lettura/scrittura primario a non essere eseguito.

Come illustrato nell'immagine seguente, la struttura di replica è contenuta in uno shard (chiamato Gruppo di nodi nell'API/CLI) che è contenuto all'interno di un cluster Redis. I cluster Redis (modalità cluster disabilitata) dispongono sempre di una partizione. I cluster Redis (modalità cluster abilitata) possono avere fino a 500 shard, con i dati partizionati tra le shard. Puoi creare un cluster con un numero più alto di partizioni e un numero più basso di repliche per un totale di 90 nodi per cluster. Questa configurazione del cluster può andare da 90 partizioni e 0 repliche a 15 partizioni e 5 repliche che è il numero massimo consentito di repliche.

Il limite del nodo o della partizione può essere aumentato fino a un massimo di 500 per cluster se la versione del motore Redis è 5.0.6 o superiore. Ad esempio, è possibile scegliere di configurare un cluster a 500 nodi che varia tra 83 partizioni (un primario e 5 repliche per partizione) e 500 partizioni (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. Per ulteriori informazioni, consulta [Creazione di un gruppo di sottoreti](#).

Per le versioni inferiori alla 5.0.6, il limite è 250 per cluster.

Per richiedere un aumento dei limiti, consulta [AWS Limiti del servizio](#) e seleziona il tipo di limite Nodi per cluster per tipo di istanza.



Il cluster Redis (modalità cluster disabilitata) dispone di una partizione e da 0 a 5 nodi di replica

Se nel cluster con repliche è abilitata la funzione Multi-AZ e il nodo primario non viene eseguito, il nodo primario esegue il failover in una replica di lettura. Poiché i dati sui nodi di replica vengono aggiornati in maniera asincrona, è possibile che alcuni dati vengano persi a causa della latenza di aggiornamento dei nodi di replica. Per ulteriori informazioni, consulta [Limitazione degli errori con Redis in esecuzione](#).

Argomenti

- [Comprendere la replica Redis](#)
- [Replica: Redis \(modalità cluster disabilitata\) e Redis \(modalità cluster abilitata\)](#)
- [Riduzione dei tempi di inattività in ElastiCache for Redis con Multi-AZ](#)
- [Modalità di implementazione di sincronizzazione e backup](#)
- [Creazione di un gruppo di replica Redis](#)
- [Visualizzazione dei dettagli del gruppo di replica](#)
- [Individuazione di endpoint del gruppo di replica](#)
- [Modifica di un gruppo di replica](#)
- [Eliminazione di un gruppo di replica](#)
- [Modifica del numero di repliche](#)
- [Promozione di una replica di lettura sul primario, per gruppi di replica Redis \(modalità cluster disabilitata\)](#)

Comprendere la replica Redis

L'implementazione della replica in Redis può essere eseguita in due modi:

- Con una singola partizione che contiene tutti i dati del cluster in ogni nodo: Redis (modalità cluster disabilitata)
- Con dati partizionati su un massimo di 500 partizioni : Redis (modalità cluster abilitata)

Ogni partizione in un gruppo di replica dispone di un singolo nodo primario lettura/scrittura e fino a 5 nodi di replica di sola lettura. Puoi creare un cluster con un numero più alto delle partizioni e un numero più basso di repliche per un totale di 90 nodi per cluster. Questa configurazione del cluster può andare da 90 partizioni e 0 repliche a 15 partizioni e 5 repliche che è il numero massimo consentito di repliche.

Il limite del nodo o della partizione può essere aumentato fino a un massimo di 500 per cluster se la versione del motore Redis è 5.0.6 o superiore. Ad esempio, è possibile scegliere di configurare un cluster a 500 nodi che varia tra 83 partizioni (un primario e 5 repliche per partizione) e 500 partizioni (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. Per ulteriori informazioni, consulta [Creazione di un gruppo di sottoreti](#).

Per le versioni inferiori alla 5.0.6, il limite è 250 per cluster.

Per richiedere un aumento dei limiti, consulta [AWS Limiti del servizio](#) e seleziona il tipo di limite Nodi per cluster per tipo di istanza.

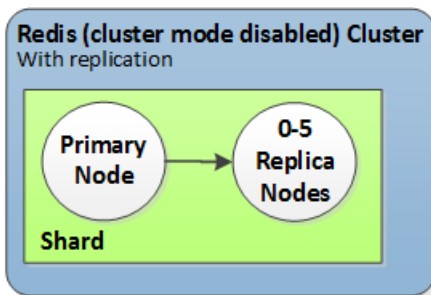
Argomenti

- [Redis \(modalità cluster disabilitata\)](#)
- [Redis \(modalità cluster abilitata\)](#)

Redis (modalità cluster disabilitata)

Un cluster Redis (modalità cluster disabilitata) dispone di una singola partizione, all'interno del quale si trova una raccolta di nodi Redis; un nodo di lettura/scrittura primario e fino a cinque nodi di sola lettura secondari. Ogni replica di lettura conserva una copia dei dati del nodo primario del cluster. Meccanismi di replica asincrona vengono utilizzati per mantenere le repliche di lettura sincronizzate

con il nodo primario. Le applicazioni possono leggere da qualsiasi nodo nel cluster. Le applicazioni possono scrivere solo nel nodo primario. Le repliche di lettura migliorano il throughput di lettura e proteggono dalla perdita di dati nei casi di mancata esecuzione del nodo.



Cluster Redis (modalità cluster disabilitata) con un singolo shard e nodi di replica

Puoi utilizzare i cluster Redis (modalità cluster disabilitata) con nodi di replica per dimensionare la soluzione Redis per ElastiCache al fine di gestire applicazioni con attività di lettura intensiva o per supportare un numero elevato di client che leggono contemporaneamente dallo stesso cluster.

Tutti i nodi in un cluster Redis (modalità cluster disabilitata) devono risiedere nella stessa regione.

Quando aggiungi una replica di lettura a un cluster, tutti i dati del nodo primario vengono copiati nel nuovo nodo. Da quel punto in poi, ogni volta che vengono scritti dati nel nodo primario, le modifiche vengono propagate in modo asincrono a tutte le repliche di lettura.

Per migliorare la tolleranza ai guasti e ridurre il tempo di inattività di scrittura, abilita la funzione Multi-AZ con failover automatico per il tuo cluster Redis (modalità cluster disabilitata) con repliche. Per ulteriori informazioni, consulta [Riduzione dei tempi di inattività in ElastiCache for Redis con Multi-AZ](#).

Puoi modificare i ruoli dei nodi all'interno del cluster Redis (modalità cluster disabilitata), con il nodo primario e una delle repliche che si scambiano i ruoli. Ciò può essere fatto per motivi di tuning delle prestazioni. Ad esempio, in un'applicazione Web con attività di scrittura elevata, puoi scegliere il nodo che presenta la latenza di rete più bassa. Per ulteriori informazioni, consulta [Promozione di una replica di lettura sul primario, per gruppi di replica Redis \(modalità cluster disabilitata\)](#).

Redis (modalità cluster abilitata)

Un cluster Redis (modalità cluster abilitata) è composto da 1 a 500 partizioni (API/CLI: gruppi di nodi). Ogni partizione dispone di un nodo primario e un massimo di cinque nodi di replica di sola lettura. La configurazione può variare da 90 partizioni e 0 repliche a 15 partizioni e 5 repliche che è il numero massimo consentito di repliche.

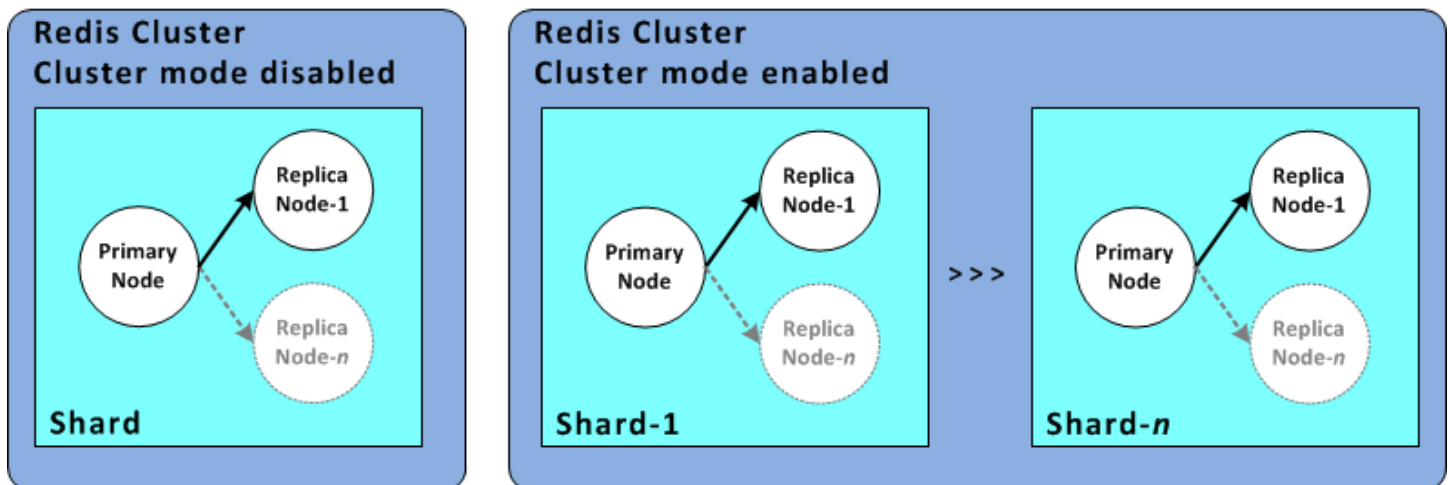
Il limite del nodo o della partizione può essere aumentato fino a un massimo di 500 per cluster se la versione del motore Redis è 5.0.6 o superiore. Ad esempio, è possibile scegliere di configurare un cluster a 500 nodi che varia tra 83 partizioni (un primario e 5 repliche per partizione) e 500 partizioni (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. Per ulteriori informazioni, consulta [Creazione di un gruppo di sottoreti](#).

Per le versioni inferiori alla 5.0.6, il limite è 250 per cluster.

Per richiedere un aumento dei limiti, consulta [AWS Limiti del servizio](#) e scegliere il tipo di limite Nodi per cluster per tipo di istanza.

Ogni replica di lettura in uno shard conserva una copia dei dati del nodo primario dello shard. Meccanismi di replica asincrona vengono utilizzati per mantenere le repliche di lettura sincronizzate con il nodo primario. Le applicazioni possono leggere da qualsiasi nodo nel cluster. Le applicazioni possono scrivere solo nei nodi primari. Le repliche di lettura migliorano la scalabilità in lettura e proteggono dalla perdita di dati. I dati vengono partizionati tra gli shard in un cluster Redis (modalità cluster abilitata).

Le applicazioni utilizzano il cluster Redis (modalità cluster abilitata) endpoint di configurazione per connettersi con i nodi del cluster. Per ulteriori informazioni, consulta [Individuazione degli endpoint di connessione](#).



Cluster Redis (modalità cluster abilitata) con più shard e nodi di replica

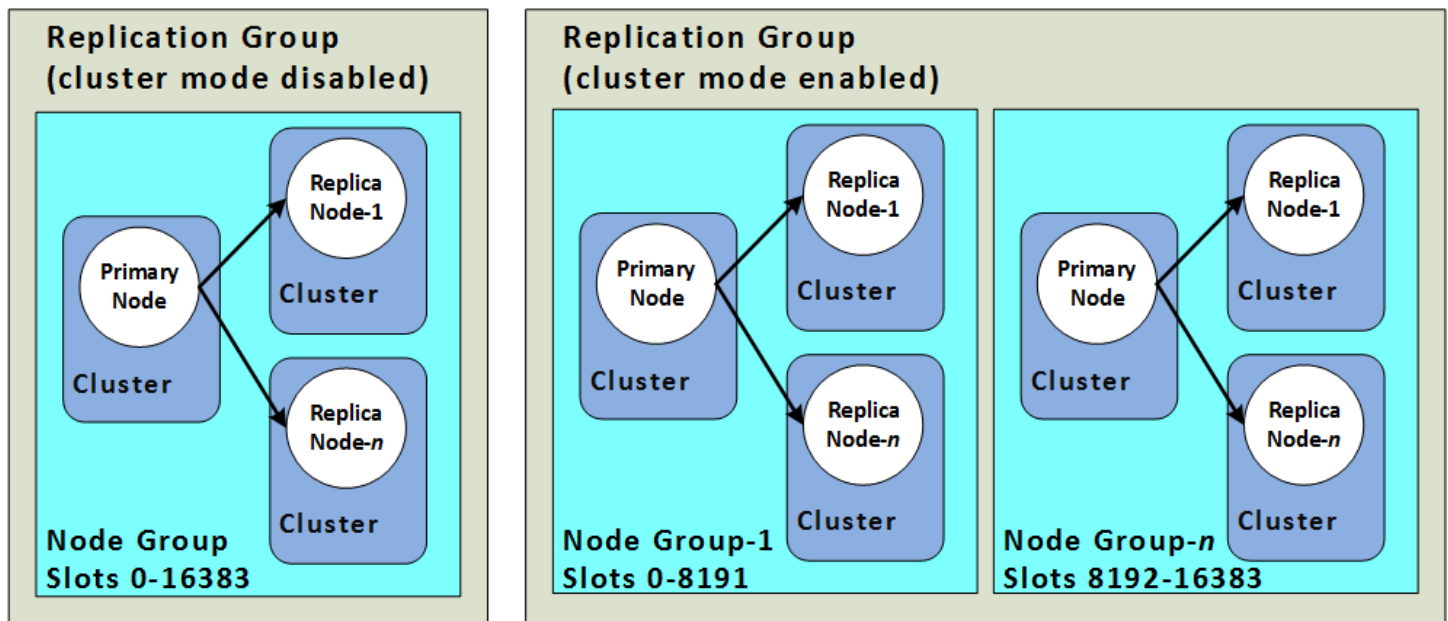
Tutti i nodi in un cluster Redis (modalità cluster abilitata) devono risiedere nella stessa regione. Per migliorare la tolleranza ai guasti, puoi eseguire il provisioning dei nodi primari e delle repliche di lettura in più zone di disponibilità all'interno di tale regione.

Attualmente, in Redis (modalità cluster abilitata), esistono alcune limitazioni.

- Non puoi promuovere manualmente nodi di replica al nodo primario.

Replica: Redis (modalità cluster disabilitata) e Redis (modalità cluster abilitata)

A partire da Redis versione 3.2, è possibile creare uno di due tipi di cluster Redis distinti (API/CLI: gruppi di replica). Un cluster Redis ((modalità cluster disabilitata) dispone sempre di un singolo shard (API/CLI: gruppo di nodi) con un massimo di 5 nodi di replica di lettura. Un cluster Redis (modalità cluster abilitata) dispone di un massimo di 500 shard ciascuno dei quali contiene da 1 a 5 nodi di replica di lettura.





Cluster Redis (modalità cluster disabilitata) e Redis (modalità cluster abilitata)

La tabella seguente riassume le differenze importanti tra cluster Redis (modalità cluster disabilitata) e Redis (modalità cluster abilitata).

Confronto tra cluster Redis (modalità cluster disabilitata) e Redis (modalità cluster abilitata)

Caratteristica	Redis (modalità cluster disabilitata)	Redis (modalità cluster abilitata)
Modificabili	Sì. Supporta l'aggiunta e l'eliminazione di nodi di	Limitato. Per ulteriori informazioni, consulta Versioni e aggiornamenti del motore e

Caratteristica	Redis (modalità cluster disabilitata)	Redis (modalità cluster abilitata)
	replica e il dimensionamento orizzontale del tipo di nodo.	Ridimensionamento dei cluster in Redis (modalità cluster attivata) .
Partizionamento dei dati	No	Sì
Shard	1	Da 1 a 500
Repliche di lettura	Da 0 a 5 <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Important Se non esistono repliche e il nodo non viene eseguito, tutti i dati vengono persi.</p> </div>	Da 0 a 5 per partizione. <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Important Se non esistono repliche e il nodo non viene eseguito, tutti i dati in tale shard vengono persi.</p> </div>
Multi-AZ	Sì, con almeno 1 replica. Facoltativo. Attivata per impostazione predefinita.	Sì Facoltativo. Attivata per impostazione predefinita.
Snapshot (Backup)	Sì, creando un singolo file .rdb.	Sì, creando un file .rdb univoco per ogni shard.
Ripristina	Sì, utilizzando un singolo file .rdb da un cluster Redis (modalità cluster disabilitata).	Sì, utilizzando file .rdb da un cluster Redis (modalità cluster disabilitata) o Redis (modalità cluster abilitata).
Supportato da	Tutte le versioni Redis	Redis 3.2 e successive

Caratteristica	Redis (modalità cluster disabilitata)	Redis (modalità cluster abilitata)
Motore aggiornabile	Sì, con alcuni limiti. Per ulteriori informazioni, consulta Versioni e aggiornamenti del motore .	Sì, con alcuni limiti. Per ulteriori informazioni, consulta Versioni e aggiornamenti del motore .
Encryption (Crittografia)	Versioni 3.2.6 (pianificata per fine vita (EOL), consultare e Pianificazione della fine del ciclo di vita delle versioni di Redis), 4.0.10 e versioni successive.	Versioni 3.2.6 (pianificata per fine vita (EOL), consultare e Pianificazione della fine del ciclo di vita delle versioni di Redis), 4.0.10 e versioni successive.
Conforme a HIPAA	Versioni 3.2.6 (pianificata per fine vita (EOL), consultare e Pianificazione della fine del ciclo di vita delle versioni di Redis), 4.0.10 e versioni successive.	Versioni 3.2.6 (pianificata per fine vita (EOL), consultare e Pianificazione della fine del ciclo di vita delle versioni di Redis), 4.0.10 e versioni successive.
Conforme a DSS PCI	Versioni 3.2.6 (pianificata per fine vita (EOL), consultare e Pianificazione della fine del ciclo di vita delle versioni di Redis), 4.0.10 e versioni successive.	Versioni 3.2.6 (pianificata per fine vita (EOL), consultare e Pianificazione della fine del ciclo di vita delle versioni di Redis), 4.0.10 e versioni successive.
Resharding online	N/D	Versione 3.2.10 (pianificata per fine vita (EOL), consultare e Pianificazione della fine del ciclo di vita delle versioni di Redis) e versioni successive.

scegliere il servizio più adatto.

Quando si sceglie tra Redis (modalità cluster disattivata) o Redis (modalità cluster abilitata), considerare i seguenti fattori:

- Dimensionamento e partizionamento - Le esigenze aziendali cambiano. Occorre fornire il provisioning per picchi di richiesta o ridimensionare quando la richiesta cambia. Redis (modalità cluster disabilitata) supporta il dimensionamento. Puoi ridimensionare la capacità di lettura aggiungendo o eliminando nodi di replica, oppure può ridimensionare la capacità passando a un tipo di nodo più grande. Entrambe queste operazioni richiedono tempo. Per ulteriori informazioni, consulta [Ridimensionamento dei cluster Redis \(modalità cluster disabilitata\) con nodi di replica](#).

Redis (modalità cluster abilitata) supporta il partizionamento dei dati fino a un massimo di 500 gruppi di nodi. Puoi modificare dinamicamente il numero di shard quando le esigenze aziendali cambiano. Un vantaggio del partizionamento è la possibilità di distribuire il carico su un numero maggiore di endpoint, riducendo i colli di bottiglia di accesso durante i picchi di richiesta. Inoltre, puoi ospitare un set di dati più grande poiché i dati possono essere distribuiti su più server. Per informazioni sul dimensionamento delle partizioni, consulta [Ridimensionamento dei cluster in Redis \(modalità cluster attivata\)](#).

- Dimensioni nodo e numero di nodi Poiché un cluster Redis (modalità cluster disabilitata) dispone di un solo shard, il tipo di nodo deve essere abbastanza grande da ospitare tutti i dati del cluster più l'overhead necessario. D'altra parte, poiché è possibile partizionare i dati tra più shard quando si utilizza un cluster Redis (modalità cluster abilitata), i tipi di nodo possono essere più piccoli, anche se ne occorre un numero maggiore. Per ulteriori informazioni, consulta [Scelta delle dimensioni dei nodi](#).
- Letture e scritture : Se il carico primario sul cluster è causato da dati di lettura delle applicazioni, puoi dimensionare un cluster Redis (modalità cluster disabilitata) aggiungendo ed eliminando repliche di lettura. Tuttavia, esiste un numero massimo di 5 repliche di lettura. Se il carico sul cluster è di scrittura, puoi trarre vantaggio dagli endpoint di scrittura aggiuntivi di un cluster Redis (modalità cluster abilitata) con più shard.

A prescindere dal tipo di cluster che scegliere di implementare, assicurati di scegliere un tipo di nodo adeguato per le esigenze correnti e future.

Riduzione dei tempi di inattività in ElastiCache for Redis con Multi-AZ

Esistono diversi casi in cui ElastiCache for Redis potrebbe essere necessario sostituire un nodo primario, tra cui alcuni tipi di manutenzione programmata e l'evento poco probabile di un fallimento in un nodo primario o zona di disponibilità.

Questa sostituzione comporta alcuni tempi di inattività per il cluster, ma se la funzione Multi-AZ è abilitata, il tempo di inattività viene ridotto al minimo. Il ruolo del nodo primario eseguirà automaticamente il failover su una delle repliche di lettura. Non è necessario creare e sottoporre a provisioning un nuovo nodo primario, poiché ElastiCache gestirà questo problema 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.

ElastiCache propaga inoltre il nome DNS (Domain Name Service) della replica promossa. Questo perché se l'applicazione scrive nell'endpoint primario, allora non è richiesta alcuna modifica dell'endpoint nell'applicazione. Tuttavia, poiché la lettura viene eseguita da singoli endpoint, occorre assicurarsi di modificare l'endpoint di lettura della replica promossa al nodo primario nel nuovo endpoint della replica.

In caso di sostituzioni dei nodi pianificati, iniziati a causa di aggiornamenti di manutenzione o aggiornamenti self-service, fare attenzione a quanto segue:

- Per ElastiCache for Redis Cluster, le sostituzioni dei nodi pianificate vengono completate mentre il cluster esegue le richieste di scrittura in entrata.
- Per i cluster Redis con Modalità cluster disabilitata con funzione Multi-AZ abilitata e in esecuzione sul motore 5.0.6 o versioni successive, le sostituzioni nodi pianificate vengono completate mentre il cluster serve le richieste di scrittura in entrata.
- Per i cluster Redis con Modalità cluster disabilitata con funzione Multi-AZ abilitata e in esecuzione sul motore 4.0.10 o versione precedente, è possibile che si verifichi una breve interruzione di scrittura, associata agli aggiornamenti DNS. Questa interruzione potrebbe richiedere fino a pochi secondi. Questo processo è molto più veloce rispetto a dover ricreare e rieseguire il provisioning di un nuovo primario, come accade nel caso in cui la funzione Multi-AZ non viene abilitata.

Puoi abilitare la funzione Multi-AZ mediante la Console di gestione ElastiCache, l'AWS CLI o l'API ElastiCache .

L'abilitazione della funzione Multi-AZ ElastiCache sul cluster Redis (nell'API e CLI, gruppo di replica) migliora la tolleranza ai guasti. Questo vale in particolare nei casi in cui il cluster principale di lettura/

scrittura del cluster diventa irraggiungibile o si guasta per qualche motivo. La funzione Multi-AZ è supportata solo su cluster Redis che hanno più di un nodo in ogni partizione.

Argomenti

- [Abilitazione della funzione Multi-AZ](#)
- [Risposte per scenari di errore relativi alla funzione Multi-AZ](#)
- [Test del failover automatico](#)
- [Limitazioni su Redis Multi-AZ](#)

Abilitazione della funzione Multi-AZ

Puoi abilitare la funzione Multi-AZ quando crei o modifichi un cluster (API o CLI, gruppo di replica) utilizzando la console ElastiCache, l' AWS CLI o l'API ElastiCache.

Puoi abilitare la funzione Multi-AZ solo su cluster Redis (modalità cluster disabilitata) in cui è abilitata almeno una replica di lettura. I cluster senza repliche di lettura non forniscono alta disponibilità o tolleranza ai guasti. Per informazioni sulla creazione di un cluster con replica, consulta [Creazione di un gruppo di replica Redis](#). Per informazioni sull'aggiunta di una replica di lettura a un cluster con replica, consulta [Aggiunta di una replica di lettura per i gruppi di replica Redis \(modalità cluster disabilitata\)](#).

Argomenti

- [Abilitazione del Multi-AZ \(console\)](#)
- [Abilitazione di Multi-AZ \(AWS CLI\)](#)
- [Abilitazione Multi-AZ \(API ElastiCache\)](#)

Abilitazione del Multi-AZ (console)

Puoi abilitare la funzione Multi-AZ utilizzando la console ElastiCache quando crei un nuovo cluster Redis o modificando un cluster Redis con replica.

La funzione Multi-AZ è abilitata per impostazione predefinita nei cluster Redis (modalità cluster abilitata).

⚠ Important

ElastiCache abilita automaticamente Multi-AZ solo se il cluster contiene almeno una replica in una zona di disponibilità diversa da quella primaria in tutte le partizioni.

Abilitazione della funzione Multi-AZ durante la creazione di un cluster ElastiCache utilizzando la console

Per ulteriori informazioni su questo processo, consulta [Creazione di un cluster Redis \(modalità cluster disabilitata\) \(Console\)](#). Assicurati di disporre di una o più repliche e di abilitare la funzione Multi-AZ.

Abilitazione della funzione Multi-AZ in un cluster esistente (console)

Per ulteriori informazioni su questo processo, consulta [Utilizzando il AWS Management Console](#) nella sezione relativa alla modifica di un cluster.

Abilitazione di Multi-AZ (AWS CLI)

Nell'esempio di codice seguente viene utilizzata l'AWS CLI per abilitare la funzione Multi-AZ per il gruppo di replica `redis12`.

⚠ Important

Il gruppo di replica `redis12` deve esistere già e disporre di almeno una replica di lettura.

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id redis12 \  
  --automatic-failover-enabled \  
  --multi-az-enabled \  
  --apply-immediately
```

Per Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id redis12 ^  
  --automatic-failover-enabled ^
```

```
--multi-az-enabled ^
--apply-immediately
```

L'output JSON di questo comando dovrebbe essere simile a quanto segue:

```
{
  "ReplicationGroup": {
    "Status": "modifying",
    "Description": "One shard, two nodes",
    "NodeGroups": [
      {
        "Status": "modifying",
        "NodeGroupMembers": [
          {
            "CurrentRole": "primary",
            "PreferredAvailabilityZone": "us-west-2b",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
              "Port": 6379,
              "Address":
"redis12-001.v5r9dc.0001.usw2.cache.amazonaws.com"
            },
            "CacheClusterId": "redis12-001"
          },
          {
            "CurrentRole": "replica",
            "PreferredAvailabilityZone": "us-west-2a",
            "CacheNodeId": "0001",
            "ReadEndpoint": {
              "Port": 6379,
              "Address":
"redis12-002.v5r9dc.0001.usw2.cache.amazonaws.com"
            },
            "CacheClusterId": "redis12-002"
          }
        ],
        "NodeGroupId": "0001",
        "PrimaryEndpoint": {
          "Port": 6379,
          "Address": "redis12.v5r9dc.ng.0001.usw2.cache.amazonaws.com"
        }
      }
    ],
  },
}
```

```
    "ReplicationGroupId": "redis12",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "enabling",
    "MultiAZ": "enabled",
    "SnapshotWindow": "07:00-08:00",
    "SnapshottingClusterId": "redis12-002",
    "MemberClusters": [
      "redis12-001",
      "redis12-002"
    ],
    "PendingModifiedValues": {}
  }
}
```

Per ulteriori informazioni, consulta questi argomenti nel Riferimento ai comandi AWS CLI:

- [create-cache-cluster](#)
- [create-replication-group](#)
- [modify-replication-group](#) nel Riferimento ai comandi AWS CLI.

Abilitazione Multi-AZ (API ElastiCache)

Nell'esempio di codice seguente viene utilizzata l'API ElastiCache per abilitare la funzione Multi-AZ per il gruppo di replica `redis12`.

Note

Per utilizzare questo esempio, il gruppo di replica `redis12` deve esistere già e disporre di almeno una replica di lettura.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyReplicationGroup
&ApplyImmediately=true
&AutoFailover=true
&MultiAZEnabled=true
&ReplicationGroupId=redis12
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
```

```
&Timestamp=20140401T192317Z  
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta questi argomenti nella Documentazione di riferimento dell'API ElastiCache:

- [CreateCacheCluster](#)
- [CreateReplicationGroup](#)
- [ModifyReplicationGroup](#)

Risposte per scenari di errore relativi alla funzione Multi-AZ

Prima dell'introduzione della funzione Multi-AZ, ElastiCache rilevava e sostituiva i nodi non riusciti del cluster ricreando ed eseguendo nuovamente il provisioning del nodo non riuscito. Abilitando la funzione Multi-AZ, un nodo primario non riuscito esegue il failover nella replica con il ritardo di replica minimo. La replica selezionata viene promossa automaticamente al nodo primario. Questa è un'operazione molto più rapida rispetto alla creazione e al provisioning di un nuovo nodo primario. Questo processo richiede in genere pochi secondi prima che sia possibile scrivere nuovamente nel cluster.

Quando la funzione Multi-AZ è abilitata, ElastiCache esegue il monitoraggio continuo dello 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 quando il nodo primario e alcune repliche di lettura non riescono](#)
- [Scenari di fallimento quando l'intero cluster non riesce](#)

Scenari di errore quando solo il nodo primario non riesce

Se il nodo primario non riesce, la replica di lettura con il tempo di replica minimo viene promossa al cluster primario. Viene quindi creata una replica di lettura sostitutiva e viene eseguito il provisioning nella stessa zona di disponibilità del nodo primario non riuscito.

Quando solo il nodo primario non riesce, ElastiCache Multi-AZ esegue le seguenti operazioni:

1. Il nodo primario non riuscito viene portato offline.
2. La replica di lettura con il tempo di replica minimo viene promossa al nodo primario.

Le scritture possono riprendere non appena il processo di promozione viene completato, in genere pochi secondi. Se l'applicazione sta scrivendo nell'endpoint primario, non occorre cambiare l'endpoint delle letture o scritture. ElastiCache propaga il nome DNS della replica promossa.

3. Una replica di lettura sostitutiva viene avviata e sottoposta a provisioning.

La replica di lettura sostitutiva viene avviata nella zona di disponibilità in cui si trovava il nodo primario non riuscito, in modo da mantenere la distribuzione dei nodi.

4. Le repliche si sincronizzano con il nuovo nodo primario.

Dopo che la nuova replica è disponibile, tieni presente questi effetti:

- Endpoint primario - Non è necessario apportare modifiche alla tua applicazione, poiché il nome DNS del nuovo nodo primario viene propagato all'endpoint primario.
- Endpoint di lettura - L'endpoint di lettura viene aggiornato automaticamente per puntare ai nuovi nodi di replica.

Per informazioni sull'individuazione degli endpoint di un cluster, consulta i seguenti argomenti:

- [Individuazione degli endpoint di un cluster Redis \(modalità cluster disabilitata\) \(console\)](#)
- [Individuazione degli endpoint per gruppi di replica \(AWS CLI\)](#)
- [Individuazione di endpoint dei gruppi di replica \(API ElastiCache\)](#)

Scenari di errore quando il nodo primario e alcune repliche di lettura non riescono

Se il nodo primario e almeno una replica di lettura non riescono, la replica disponibile con il ritardo di replica minimo viene promossa al cluster primario. Nuove repliche di lettura vengono inoltre create e sottoposte a provisioning nelle stesse zone di disponibilità dei nodi non riusciti e della replica che è stata promossa al nodo primario.

Quando il nodo primario e alcune repliche di lettura non riescono, ElastiCache Multi-AZ esegue le seguenti operazioni:

1. Il nodo primario non riuscito e le repliche di lettura non riuscite vengono portate offline.
2. La replica disponibile con il tempo di replica minimo viene promossa al nodo primario.

Le scritture possono riprendere non appena il processo di promozione viene completato, in genere pochi secondi. Se l'applicazione sta scrivendo nell'endpoint primario, non occorre cambiare l'endpoint delle scritture. ElastiCache propaga il nome DNS della replica promossa.

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 cluster si sincronizzano con il nuovo nodo primario.

Dopo che i nuovi nodi sono disponibili, occorre apportare le seguenti modifiche all'applicazione:

- Endpoint primario - Non apportare modifiche all'applicazione. Il nome DNS del nuovo nodo primario viene propagato all'endpoint primario.
- Endpoint di lettura - L'endpoint di lettura viene aggiornato automaticamente per puntare ai nuovi nodi di replica.

Per informazioni sull'individuazione degli endpoint di un gruppo di replica, consulta i seguenti argomenti:

- [Individuazione degli endpoint di un cluster Redis \(modalità cluster disabilitata\) \(console\)](#)
- [Individuazione degli endpoint per gruppi di replica \(AWS CLI\)](#)
- [Individuazione di endpoint dei gruppi di replica \(API ElastiCache\)](#)

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.

In questo scenario, tutti i dati nel cluster vengono persi a causa del guasto di ogni nodo nel cluster. Questa eventualità è rara.

Quando l'intero cluster non riesce, ElastiCache Multi-AZ esegue le seguenti operazioni:

1. Il nodo primario e le repliche di lettura non riusciti vengono portati offline.
2. Un nodo primario sostitutivo viene creato e sottoposto a provisioning.
3. Repliche sostitutive vengono create e sottoposte a provisioning.

Le sostituzioni vengono create nelle zone di disponibilità dei nodi non riusciti, in modo da mantenere la distribuzione dei nodi.

Poiché l'intero cluster non è riuscito, i dati vengono persi e tutti i nuovi nodi vengono avviati come inattivi.

Poiché ciascuno dei nodi sostitutivi ha lo stesso endpoint del nodo che sta sostituendo, non occorre apportare modifiche all'endpoint nell'applicazione.

Per informazioni sull'individuazione degli endpoint di un gruppo di replica, consulta i seguenti argomenti:

- [Individuazione degli endpoint di un cluster Redis \(modalità cluster disabilitata\) \(console\)](#)
- [Individuazione degli endpoint per gruppi di replica \(AWS CLI\)](#)
- [Individuazione di endpoint dei gruppi di replica \(API ElastiCache\)](#)

Ti consigliamo di creare il nodo primario e le repliche di lettura in zone di disponibilità diverse per aumentare il livello di tolleranza ai guasti.

Test del failover automatico

Dopo aver abilitato la funzione di failover automatico, puoi testarla utilizzando la console , la AWS CLI e l'API ElastiCache.

Durante il test, tieni presente quanto segue:

- Puoi utilizzare questa operazione per testare il failover automatico su un massimo di cinque partizioni (chiamati gruppi di nodi nell'API ElastiCache e inAWS CLI) in qualsiasi periodo continuo di 24 ore.
- Se chiami questa operazione su partizioni in cluster diversi (chiamati gruppi di replica nell'API e in CLI), puoi eseguire le chiamate contemporaneamente.
- In alcuni casi, puoi chiamare questa operazione più volte su partizioni diversi nello stesso gruppo di replica Redis (modalità cluster abilitata) 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 terminata, puoi controllare gli eventi utilizzando la console Amazon ElastiCache, l' AWS CLI o l'API ElastiCache. Cerca i seguenti eventi correlati al failover automatico, elencati qui in ordine di occorrenza:
 1. Messaggio del gruppo di replica: `Test Failover API called for node group <node-group-id>`
 2. Messaggio del cluster di cache: `Failover from primary node <primary-node-id> to replica node <node-id> completed`
 3. Messaggio del gruppo di replica: `Failover from primary node <primary-node-id> to replica node <node-id> completed`
 4. Messaggio del cluster di cache: `Recovering cache nodes <node-id>`
 5. Messaggio del cluster di cache: `Finished recovery for cache nodes <node-id>`

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Visualizzazione di eventi ElastiCache C](#) nella Guida per l'utente ElastiCache
- [DescribeEvents](#) nella Documentazione di riferimento dell'API ElastiCache
- [describe-events](#) nel Riferimento ai comandi AWS CLI.
- Questa API è progettata per testare il comportamento dell'applicazione in caso di failover di ElastiCache. 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

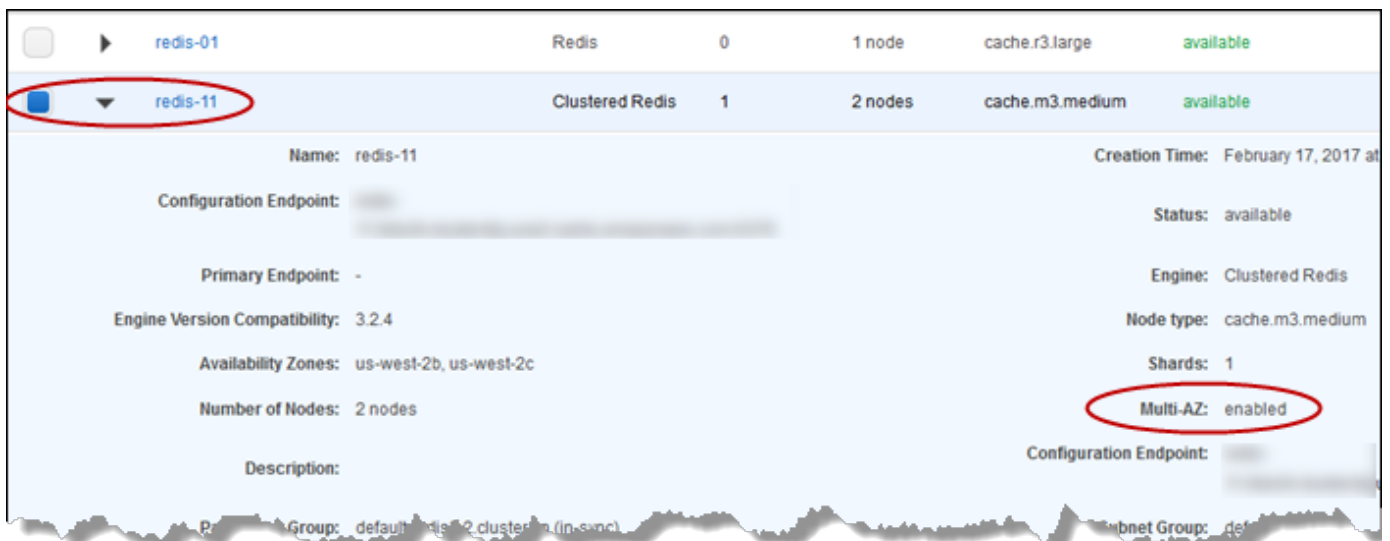
- [Test del failover automatico utilizzando la AWS Management Console](#)
- [Test del failover automatico utilizzando la AWS CLI](#)
- [Test del failover automatico mediante l'API ElastiCache](#)

Test del failover automatico utilizzando la AWS Management Console

Utilizza la procedura seguente per testare il failover automatico con la console.

Per testare il failover automatico

1. Accedi alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione, scegli Redis.
3. Dall'elenco di cluster Redis, scegli la casella a sinistra del cluster che desideri testare. Questo cluster deve disporre almeno di un nodo di replica di lettura.
4. 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 [Utilizzando il AWS Management Console](#).



5. Per Redis (modalità cluster disabilitata), scegliere il nome del cluster.

Per Redis (modalità cluster abilitata), si applica quanto segue:

- a. Seleziona il nome del cluster.

- b. Nella pagina Shards (Partizioni), per la partizione (chiamato gruppo di nodi nell'API e in CLI) su cui desideri testare il failover, scegliere il nome della partizione.
6. Nella pagina dei nodi, scegli Failover Primary (Failover primario).
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 (Test Failover API called) e completato (Recovery completed).

Test del failover automatico utilizzando la AWS CLI

Puoi testare il failover automatico su qualsiasi cluster abilitato per la funzione Multi-AZ mediante l'operazione `test-failover` di AWS CLI.

Parametri

- `--replication-group-id` – Obbligatorio. Il gruppo di replica (sulla console, cluster) da testare.
 - `--node-group-id` – Obbligatorio. Il nome del gruppo di nodi sui cui testare il failover automatico.
- Puoi testare un massimo di cinque gruppi di nodi in un periodo continuo di 24 ore.

L'esempio seguente utilizza AWS CLI per testare il failover automatico sul gruppo di nodi `redis00-0003` nel cluster Redis (modalità cluster abilitata) `redis00`.

Example Test del failover automatico

Per Linux, macOS o Unix:

```
aws elasticache test-failover \  
  --replication-group-id redis00 \  
  --node-group-id redis00-0003
```

Per Windows:

```
aws elasticache test-failover ^
```

```
--replication-group-id redis00 ^  
--node-group-id redis00-0003
```

L'aspetto dell'output del comando precedente è simile al seguente.

```
{  
  "ReplicationGroup": {  
    "Status": "available",  
    "Description": "1 shard, 3 nodes (1 + 2 replicas)",  
    "NodeGroups": [  
      {  
        "Status": "available",  
        "NodeGroupMembers": [  
          {  
            "CurrentRole": "primary",  
            "PreferredAvailabilityZone": "us-west-2c",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,  
              "Address":  
"redis1x3-001.7ekv3t.0001.usw2.cache.amazonaws.com"  
            },  
            "CacheClusterId": "redis1x3-001"  
          },  
          {  
            "CurrentRole": "replica",  
            "PreferredAvailabilityZone": "us-west-2a",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,  
              "Address":  
"redis1x3-002.7ekv3t.0001.usw2.cache.amazonaws.com"  
            },  
            "CacheClusterId": "redis1x3-002"  
          },  
          {  
            "CurrentRole": "replica",  
            "PreferredAvailabilityZone": "us-west-2b",  
            "CacheNodeId": "0001",  
            "ReadEndpoint": {  
              "Port": 6379,  
              "Address":  
"redis1x3-003.7ekv3t.0001.usw2.cache.amazonaws.com"  
            }  
          }  
        ]  
      }  
    ]  
  }  
}
```

```

        },
        "CacheClusterId": "redis1x3-003"
    }
],
"NodeGroupId": "0001",
"PrimaryEndpoint": {
    "Port": 6379,
    "Address": "redis1x3.7ekv3t.ng.0001.usw2.cache.amazonaws.com"
}
}
],
"ClusterEnabled": false,
"ReplicationGroupId": "redis1x3",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "enabled",
"MultiAZ": "enabled",
"SnapshotWindow": "11:30-12:30",
"SnapshottingClusterId": "redis1x3-002",
"MemberClusters": [
    "redis1x3-001",
    "redis1x3-002",
    "redis1x3-003"
],
"CacheNodeType": "cache.m3.medium",
"DataTiering": "disabled",
"PendingModifiedValues": {}
}
}

```

Per monitorare l'avanzamento del failover, utilizza l'operazione `describe-events` di AWS CLI.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [test-failover](#) nel Riferimento ai comandi AWS CLI.
- [describe-events](#) nel Riferimento ai comandi AWS CLI.

Test del failover automatico mediante l'API ElastiCache

Puoi testare il failover automatico su qualsiasi cluster abilitato per la funzione Multi-AZ mediante l'operazione ElastiCache dell'API `TestFailover`.

Parametri

- `ReplicationGroupId` – Obbligatorio. Il gruppo di replica (sulla console, cluster) da testare.
- `NodeGroupId` – Obbligatorio. Il nome del gruppo di nodi sui cui testare il failover automatico. Puoi testare un massimo di cinque gruppi di nodi in un periodo continuo di 24 ore.

L'esempio seguente esegue il test del failover automatico sul gruppo di nodi `redis00-0003` nel gruppo di replica (sulla console, cluster) `redis00`.

Example Test del failover automatico

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=TestFailover  
&NodeGroupId=redis00-0003  
&ReplicationGroupId=redis00  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20140401T192317Z  
&X-Amz-Credential=<credential>
```

Per monitorare l'avanzamento del failover, utilizza l'operazione `DescribeEvents` dell'API ElastiCache.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [TestFailover](#) nella Documentazione di riferimento dell'API ElastiCache
- [DescribeEvents \(Descrivi eventi\)](#) nella Documentazione di riferimento dell'API ElastiCache

Limitazioni su Redis Multi-AZ


Tieni presente le seguenti limitazioni per la funzione Redis Multi-AZ:

- La funzione Multi-AZ è supportata su Redis versione 2.8.6 e successive.
- La funzione Redis Multi-AZ non è supportata sui tipi di nodi T1.
- La replica Redis è asincrona. Pertanto, quando un nodo primario esegue il failover in una replica, si può verificare una piccola perdita di dati a causa del ritardo di replica.

Quando si sceglie la replica da promuovere a nodo primario, ElastiCache for Redis sceglie quella con il ritardo di replica minimo. ovvero quella più attuale. Ciò consente di ridurre al minimo la quantità di dati persi. La replica con il ritardo di replica minimo si può trovare nella stessa zona di disponibilità del nodo primario con errore o in una zona diversa.

- Quando promuovi manualmente una replica di lettura a nodo primario su Redis (modalità cluster disabilitata), è possibile farlo solo quando la funzione Multi-AZ e la funzione di failover automatico sono disabilitate. Per promuovere una replica di lettura a nodo primario, procedi come indicato di seguito:
 1. Disabilitare la funzione Multi-AZ nel cluster.
 2. Disabilitare il failover automatico nel cluster. A questo scopo, utilizza la console Redis deselezionando la casella di controllo Failover automatico per il gruppo di replica. A tale scopo, è possibile utilizzare il comando AWS CLI impostando la proprietà `AutomaticFailoverEnabled` a `false` quando si chiama l'operazione `ModifyReplicationGroup`.
 3. Promuovere la replica di lettura al nodo primario.
 4. Riattivare la funzione Multi-AZ.
- Le funzioni ElastiCache for Redis Multi-AZ e AOF (append-only file) sono mutuamente esclusive. Se si abilita una, non si può abilitare l'altra.
- Un guasto del nodo può essere causato dal raro evento di errore di un'intera zona di disponibilità. In questo caso, la replica che sostituisce il nodo primario in errore viene creata solo quando la zona di disponibilità viene ripristinata. Ad esempio, considerare un gruppo di replica con il nodo primario in AZ e le repliche in AZ-b e AZ-c. Se il nodo primario non riesce, la replica con il tempo di replica minimo viene promossa al cluster primario. Quindi, ElastiCache crea una nuova replica in AZ-a (in cui si trovava il nodo primario non riuscito) solo quando il backup di AZ-a viene eseguito ed è disponibile.
- Un riavvio di un nodo primario iniziato dal cliente non attiva il failover automatico. Altri riavvi ed guasti attivano il failover automatico.
- Ogni volta che il nodo primario viene riavviato, i dati vengono cancellati quando ritorna online. Quando le repliche di lettura rilevano il cluster primario cancellato, cancellano la loro copia dei dati causando perdita dei dati.
- Dopo che una replica di lettura è stata promossa, le altre repliche si sincronizzano con il nuovo nodo primario. Dopo la sincronizzazione iniziale, il contenuto delle repliche viene eliminato e vengono sincronizzati i dati dal nuovo nodo primario. Questo processo di sincronizzazione causa una breve interruzione, durante la quale le repliche non sono accessibili. Il processo di

sincronizzazione causa anche un incremento di carico temporaneo sul nodo primario durante la sincronizzazione con le repliche. Questo comportamento è tipico di Redis e non è esclusivo di ElastiCache Multi-AZ. Per ulteriori informazioni su questo comportamento di Redis, consulta [Replica](#) sul sito Web di Redis.

 Important

Per Redis versione 2.8.22 e successive, non è possibile creare repliche esterne. Per le versioni Redis precedenti alla 2.8.22, si consiglia di non connettere una replica Redis esterna a un cluster ElastiCache for Redis con abilitata la funzione Multi-AZ. Questa è una configurazione non supportata che può creare problemi che impediscono a ElastiCache di eseguire correttamente il failover e il ripristino. Per connettere una replica Redis esterna a un cluster ElastiCache, verificare che la funzione Multi-AZ sia disabilitata prima di effettuare la connessione.

Modalità di implementazione di sincronizzazione e backup

Tutte le versioni supportate di Redis supportano il backup e la sincronizzazione tra i nodi primari e di replica. Tuttavia, la modalità di implementazione di backup e sincronizzazione varia in base alla versione di Redis.

Redis versione 2.8.22 e successive

La replica Redis, nelle versioni 2.8.22 e successive, scegliere tra due metodi. Per ulteriori informazioni, consulta [Versioni Redis precedenti a 2.8.22](#) e [Snapshot e ripristino](#).

Durante il processo senza fork, se i carichi di scrittura sono intensi, le scritture nel cluster vengono ritardate per garantire che non vengano accumulate troppe modifiche che impediscono la riuscita di una snapshot.

Versioni Redis precedenti a 2.8.22

Il backup e la sincronizzazione Redis in versioni precedenti alla 2.8.22 è un processo in tre fasi.

1. Fork e il processo in background serializzano i dati del cluster su disco. Questo consente di creare una snapshot point-in-time.
2. In primo piano, accumulare un change log nel buffer di output client.

Important

Se il change log supera le dimensioni del buffer di output del client, l'operazione di backup o sincronizzazione non riesce. Per ulteriori informazioni, consulta [Verifica della disponibilità di memoria sufficiente per creare uno snapshot Redis](#).

3. Infine, trasmettere i dati di cache e il change log al nodo di replica.

Creazione di un gruppo di replica Redis

Per creare un cluster con nodi di replica sono disponibili le seguenti opzioni. Una si applica quando già disponi di un cluster Redis (modalità cluster disabilitata) non associato a nessun cluster che abbia repliche da utilizzare come nodo primario. L'altra si applica quando desideri creare un nodo primario con il cluster e le repliche di lettura. Attualmente, un cluster Redis (modalità cluster abilitata) deve essere creato da zero.

Opzione 1:[Creazione di un gruppo di replica utilizzando un cluster Redis \(modalità cluster disabilitata\) disponibile](#)

Utilizza questa opzione per sfruttare un cluster Redis (modalità cluster disabilitata) a nodo singolo esistente. Specifica questo nodo esistente come il nodo primario nel nuovo cluster e quindi aggiungi individualmente da 1 a 5 repliche di lettura al cluster. Se il cluster esistente è attivo, le repliche di lettura si sincronizzano con il cluster mentre vengono create. Per informazioni, consultare [Creazione di un gruppo di replica utilizzando un cluster Redis \(modalità cluster disabilitata\) disponibile](#).

Important

Non puoi creare un cluster Redis (modalità cluster abilitata) utilizzando un cluster esistente. Per creare un cluster Redis (modalità cluster abilitata) (API/CLI:gruppo di replica), utilizzando la console ElastiCache, consulta [Creazione di un cluster Redis \(modalità cluster abilitata\) \(console\)](#).

Opzione 2:[Creazione di un gruppo di replica Redis da zero](#)

Utilizza questa opzione se non disponi già di un cluster Redis (modalità cluster disabilitata) da utilizzare come il nodo primario del cluster o se desideri creare un cluster Redis (modalità cluster disabilitata). Per informazioni, consultare [Creazione di un gruppo di replica Redis da zero](#).

Creazione di un gruppo di replica utilizzando un cluster Redis (modalità cluster disabilitata) disponibile

Un cluster disponibile è un cluster Redis a nodo singolo esistente. Attualmente, Redis (modalità cluster abilitata) non supporta la creazione di un cluster con repliche utilizzando un cluster a nodo singolo disponibile. Se desideri creare un cluster Redis (modalità cluster abilitata), consulta [Creazione di un cluster Redis \(modalità cluster abilitata\) \(console\)](#).

La seguente procedura può essere utilizzata solo se disponi di un cluster Redis (modalità cluster disabilitata) a nodo singolo. Il nodo di questo cluster diventa il nodo primario nel nuovo cluster. Se non disponi di un cluster Redis (modalità cluster disabilitata) che puoi utilizzare come il nodo primario del nuovo cluster, consulta [Creazione di un gruppo di replica Redis da zero](#).

Creazione di un gruppo di replica utilizzando un cluster Redis disponibile (console)

Consulta l'argomento [Utilizzo di AWS Management Console](#)

Creazione di un gruppo di replica utilizzando un cluster di cache Redis disponibile (AWS CLI)

La creazione di un gruppo di replica con repliche di lettura durante l'utilizzo di un cluster di cache Redis disponibile per il nodo primario quando si utilizza AWS CLI prevede due fasi.

Se utilizzi AWS CLI, puoi creare un gruppo di replica specificando il nodo autonomo disponibile come il nodo primario del cluster, `--primary-cluster-id` e il numero di nodi desiderati nel cluster utilizzando il comando CLI, `create-replication-group`. Includi i seguenti parametri.

`--replication-group-id`

Il nome del gruppo di replica che stai creando. Il valore di questo parametro viene utilizzato come la base per i nomi dei nodi aggiunti con un numero di tre cifre sequenziale aggiunto alla fine di `--replication-group-id`. Ad esempio, `sample-repl-group-001`.

I vincoli di denominazione dei gruppi di replica Redis (modalità cluster disabilitata) 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.

`--replication-group-description`

Descrizione del gruppo di replica.

--num-node-groups

Numero di nodi desiderati in questo cluster. Questo valore include il nodo primario. Il valore massimo di questo parametro è sei.

--primary-cluster-id

Il nome del nodo del cluster Redis (modalità cluster disabilitata) disponibile che deve essere impostato come nodo primario in questo gruppo di replica.

Il seguente comando consente di creare il gruppo di replica `sample-repl-group` utilizzando il cluster Redis `redis01` (modalità cluster disabilitata) disponibile come nodo primario del gruppo di replica. Vengono creati due nuovi nodi che sono repliche di lettura. Le impostazioni di `redis01` (ovvero, gruppo di parametri, gruppo di sicurezza, tipo di nodo, versione motore e così via) verranno applicate a tutti i nodi nel gruppo di replica.

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id sample-repl-group \  
  --replication-group-description "demo cluster with replicas" \  
  --num-cache-clusters 3 \  
  --primary-cluster-id redis01
```

Per Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --replication-group-description "demo cluster with replicas" ^  
  --num-cache-clusters 3 ^  
  --primary-cluster-id redis01
```

Per informazioni aggiuntive e i parametri che potrebbe essere necessario utilizzare, consulta l'argomento di AWS CLI [create-replication-group](#).

Aggiunta di repliche di lettura al gruppo di replica

Dopo che il gruppo di replica è stato creato, aggiungi da una a cinque repliche di lettura utilizzando il comando `create-cache-cluster`, assicurandoti di includere i seguenti parametri.

--cache-cluster-id

Il nome del cluster che stai aggiungendo al gruppo di replica.

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.

--replication-group-id

Il nome del gruppo di replica cui stai aggiungendo questo cluster di cache.

Ripeti questo comando per ogni replica di lettura che desideri aggiungere al gruppo di replica, modificando solo il valore del parametro `--cache-cluster-id`.

Note

Ricorda che un gruppo di replica non può contenere più di cinque repliche di lettura. Se tenti di aggiungere una replica di lettura a un gruppo di replica che contiene già cinque repliche di lettura, l'operazione non riesce.

Il codice seguente consente di aggiungere la replica di lettura `my-replica01` al gruppo di replica `sample-repl-group`. Le impostazioni del cluster primario, ovvero gruppo di parametri, gruppo di sicurezza, tipo di nodo e così via - verranno applicati ai nodi quando vengono aggiunti al gruppo di replica.

Per Linux, macOS o Unix:

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id my-replica01 \  
  --replication-group-id sample-repl-group
```

Per Windows:

```
aws elasticache create-cache-cluster ^  
  --cache-cluster-id my-replica01 ^
```

```
--replication-group-id sample-repl-group
```

L'aspetto dell'output di questo comando sarà simile al seguente.

```
{
  "ReplicationGroup": {
    "Status": "creating",
    "Description": "demo cluster with replicas",
    "ClusterEnabled": false,
    "ReplicationGroupId": "sample-repl-group",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "disabled",
    "SnapshotWindow": "00:00-01:00",
    "SnapshottingClusterId": "redis01",
    "MemberClusters": [
      "sample-repl-group-001",
      "sample-repl-group-002",
      "redis01"
    ],
    "CacheNodeType": "cache.m4.large",
    "DataTiering": "disabled",
    "PendingModifiedValues": {}
  }
}
```

Per ulteriori informazioni, consulta gli argomenti AWS CLI:

- [create-replication-group](#)
- [modify-replication-group](#)

Aggiunta di repliche a un cluster Redis standalone (modalità cluster disabilitata) (API ElastiCache)

Quando utilizzi l'API ElastiCache, puoi creare un gruppo di replica specificando il nodo autonomo disponibile come il nodo primario del cluster, `PrimaryClusterId` e il numero di nodi desiderati nel cluster utilizzando il comando CLI, `CreateReplicationGroup`. Includi i seguenti parametri.

`ReplicationGroupId`

Il nome del gruppo di replica che stai creando. Il valore di questo parametro viene utilizzato come la base per i nomi dei nodi aggiunti con un numero di tre cifre sequenziale aggiunto alla fine di `ReplicationGroupId`. Ad esempio, `sample-repl-group-001`.

I vincoli di denominazione dei gruppi di replica Redis (modalità cluster disabilitata) 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.

ReplicationGroupDescription

Descrizione del cluster con repliche.

NumCacheClusters

Numero di nodi desiderati in questo cluster. Questo valore include il nodo primario. Il valore massimo di questo parametro è sei.

PrimaryClusterId

Il nome del cluster Redis (modalità cluster disabilitata) disponibile che deve essere impostato come il nodo primario in questo cluster.

Il seguente comando consente di creare il cluster con repliche `sample-repl-group` utilizzando il cluster Redis (modalità cluster disabilitata) `redis01` disponibile come il nodo primario del gruppo di replica. Vengono creati due nuovi nodi che sono repliche di lettura. Le impostazioni di `redis01` (ovvero, gruppo di parametri, gruppo di sicurezza, tipo di nodo, versione motore e così via) verranno applicate a tutti i nodi nel gruppo di replica.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateReplicationGroup  
&Engine=redis  
&EngineVersion=6.0  
&ReplicationGroupDescription=Demo%20cluster%20with%20replicas  
&ReplicationGroupId=sample-repl-group  
&PrimaryClusterId=redis01  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta gli argomenti APL ElastiCache:

- [CreateReplicationGroup](#)
- [ModifyReplicationGroup](#)

Aggiunta di repliche di lettura al gruppo di replica

Dopo che il gruppo di replica è stato creato, aggiungi a esso da una a cinque repliche di lettura utilizzando l'operazione `CreateCacheCluster`, assicurandoti di includere i seguenti parametri.

CacheClusterId

Il nome del cluster che stai aggiungendo al gruppo di replica.

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.

ReplicationGroupId

Il nome del gruppo di replica cui stai aggiungendo questo cluster di cache.

Ripeti questa operazione per ogni replica di lettura che desideri aggiungere al gruppo di replica, modificando solo il valore del parametro `CacheClusterId`.

Il seguente codice aggiunge la replica di lettura `myReplica01` al gruppo di replica `myRep1Group`. Le impostazioni del cluster primario, ovvero gruppo di parametri, gruppo di sicurezza, tipo di nodo e così via - verranno applicati ai nodi quando vengono aggiunti al gruppo di replica.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateCacheCluster  
&CacheClusterId=myReplica01  
&ReplicationGroupId=myRep1Group  
&SignatureMethod=HmacSHA256  
&SignatureVersion=4  
&Version=2015-02-02  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Credential=[your-access-key-id]/20150202/us-west-2/elasticache/aws4_request  
&X-Amz-Date=20150202T170651Z
```

```
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date  
&X-Amz-Signature=[signature-value]
```

Per informazioni aggiuntive e i parametri da utilizzare, consulta l'argomento API ElastiCache [CreateCacheCluster](#).

Creazione di un gruppo di replica Redis da zero

Di seguito viene descritto come creare un gruppo di replica Redis senza utilizzare un cluster Redis esistente come primario. Puoi creare un gruppo di repliche Redis (modalità cluster disabilitata) o Redis (modalità cluster abilitata) utilizzando la console ElastiCache, lo AWS CLI o l'API ElastiCache.

Prima di continuare, scegliere se creare un gruppo di replica Redis (modalità cluster disabilitata) o Redis (modalità cluster abilitata). Per le linee guida, consulta [Replica: Redis \(modalità cluster disabilitata\)](#) e [Redis \(modalità cluster abilitata\)](#).

Argomenti

- [Creazione di un gruppo di replica Redis \(modalità cluster disabilitata\) da zero.](#)
- [Creazione da zero di un gruppo di replica in Redis \(modalità cluster abilitata\)](#)

Creazione di un gruppo di replica Redis (modalità cluster disabilitata) da zero.

Puoi creare un gruppo di replica Redis (modalità cluster disabilitata) da zero utilizzando la console ElastiCache, la AWS CLI o l'API ElastiCache. Un gruppo di repliche Redis (modalità cluster disabilitata) è costituito sempre da un gruppo di nodi, un cluster primario e fino a cinque repliche di lettura. I gruppi di replica Redis (modalità cluster disabilitata) non supportano il partizionamento dei dati.

Note

Il limite dei nodi/partizioni può essere aumentato fino a un massimo di 500 per ogni cluster. Per richiedere un aumento dei limiti, consulta [AWS Limite dei servizi](#) e includi il tipo di istanza nella richiesta.

Per creare un gruppo di replica Redis (modalità cluster disabilitata) da zero, adotta uno dei seguenti approcci:

Creazione da zero di un gruppo di replica Redis (modalità cluster disabilitata) (AWS CLI)

La seguente procedura consente di creare un gruppo di replica Redis (modalità cluster disabilitata) utilizzando il metodo AWS CLI.

Quando crei un gruppo di replica Redis (modalità cluster disabilitata) da zero, il gruppo di replica e tutti i relativi nodi vengono creati con una singola chiamata al comando di AWS CLI `create-replication-group`. Includi i seguenti parametri.

`--replication-group-id`

Il nome del gruppo di replica che stai creando.

I vincoli di denominazione dei gruppi di replica Redis (modalità cluster disabilitata) 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.

`--replication-group-description`

Descrizione del gruppo di replica.

--num-cache-clusters

Il numero di nodi che devono essere creati con questo gruppo di replica, nodo primario e repliche di lettura combinati.

Se abiliti Multi-AZ (--automatic-failover-enabled), il valore di --num-cache-clusters deve essere almeno 2.

--cache-node-type

Il tipo di nodo per ogni nodo nel gruppo di replica.

ElastiCache supporta i seguenti tipi di nodi. In generale, le generazioni attuali offrono più memoria e più potenza di elaborazione a un costo inferiore rispetto ai prodotti omologhi della generazione precedente.

Per ulteriori informazioni sui dettagli sulle prestazioni per ogni tipo di nodo, consulta [Tipi di istanza di Amazon EC2](#).

--data-tiering-abilitato

Impostare questo parametro se si utilizza un tipo di nodo r6gd. Se non si desidera il tiering di dati, impostare --no-data-tiering-enabled. Per ulteriori informazioni, consulta [Tiering di dati](#).

--cache-parameter-group

Specifica un gruppo di parametri che corrisponde alla versione motore. Se stai eseguendo Redis 3.2.4 o versioni successive, specifica il gruppo di parametri default.redis3.2 o un gruppo di parametri derivato da default.redis3.2 per creare un gruppo di replica Redis (modalità cluster disabilitata). Per ulteriori informazioni, consulta [Parametri specifici di Redis](#).

--network-type

ipv4, ipv6 o dual-stack. Se scegli dual-stack, devi impostare il parametro --IpDiscovery su ipv4 o ipv6.

--engine

redis

--engine-version

Per disporre del più ampio set di caratteristiche, scegliere la versione motore più recente.

I nomi dei nodi verranno derivati dal nome del gruppo di replica aggiungendo `-00#`. Ad esempio, utilizzando il nome del gruppo di replica `myReplGroup`, il nome del primario sarà `myReplGroup-001` e le repliche di lettura da `myReplGroup-002` a `myReplGroup-006`.

Se desideri abilitare la crittografia dei dati in transito o dei dati inattivi su questo gruppo di replica, aggiungi uno o entrambi i parametri `--transit-encryption-enabled` o `--at-rest-encryption-enabled` e soddisfa le seguenti condizioni.

- Il gruppo di replica deve eseguire Redis versione 3.2.6 o 4.0.10.
- Il gruppo di replica deve essere creato in un Amazon VPC.
- Devi anche includere il parametro `--cache-subnet-group`.
- Deve anche includere il parametro `--auth-token` con il valore stringa specificato dal cliente per il token AUTH (password) necessario per eseguire operazioni su questo gruppo di replica.

La seguente operazione consente di creare un gruppo di replica Redis (modalità cluster disabilitata) `sample-repl-group` con tre nodi, un nodo primario e due nodi di replica.

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id sample-repl-group \  
  --replication-group-description "Demo cluster with replicas" \  
  --num-cache-clusters 3 \  
  --cache-node-type cache.m4.large \  
  --engine redis
```

Per Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --replication-group-description "Demo cluster with replicas" ^  
  --num-cache-clusters 3 ^  
  --cache-node-type cache.m4.large ^  
  --engine redis
```

L'aspetto dell'output di questo comando è simile al seguente.

```
{
```

```
"ReplicationGroup": {
  "Status": "creating",
  "Description": "Demo cluster with replicas",
  "ClusterEnabled": false,
  "ReplicationGroupId": "sample-repl-group",
  "SnapshotRetentionLimit": 0,
  "AutomaticFailover": "disabled",
  "SnapshotWindow": "01:30-02:30",
  "MemberClusters": [
    "sample-repl-group-001",
    "sample-repl-group-002",
    "sample-repl-group-003"
  ],
  "CacheNodeType": "cache.m4.large",
  "DataTiering": "disabled",
  "PendingModifiedValues": {}
}
```

Per informazioni aggiuntive e i parametri che potrebbe essere necessario utilizzare, consulta l'argomento di AWS CLI [create-replication-group](#).

Creazione di un gruppo di replica Redis (modalità cluster disabilitata) da zero (API ElastiCache)

La seguente procedura consente di creare un gruppo di replica Redis (modalità cluster disabilitata) utilizzando l'API ElastiCache.

Quando crei un gruppo di replica Redis (modalità cluster disabilitata) da zero, il gruppo di replica e tutti i relativi nodi vengono creati con una singola chiamata all'operazione API `CreateReplicationGroup` ElastiCache . Includi i seguenti parametri.

ReplicationGroupId

Il nome del gruppo di replica che stai creando.

I vincoli di denominazione dei gruppi di replica Redis (modalità cluster abilitata) 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.

ReplicationGroupDescription

La descrizione del gruppo di replica.

NumCacheClusters

Il numero totale di nodi che devono essere creati con questo gruppo di replica, nodo primario e repliche di lettura combinati.

Se abiliti Multi-AZ (`AutomaticFailoverEnabled=true`), il valore di `NumCacheClusters` deve essere almeno 2.

CacheNodeType

Il tipo di nodo per ogni nodo nel gruppo di replica.

ElastiCache supporta i seguenti tipi di nodi. In generale, le generazioni attuali offrono più memoria e più potenza di elaborazione a un costo inferiore rispetto ai prodotti omologhi della generazione precedente.

Per ulteriori informazioni sui dettagli sulle prestazioni per ogni tipo di nodo, consulta [Tipi di istanza di Amazon EC2](#).

--data-tiering-abilitato

Impostare questo parametro se si utilizza un tipo di nodo `r6gd`. Se non si desidera il tiering di dati, impostare `--no-data-tiering-enabled`. Per ulteriori informazioni, consulta [Tiering di dati](#).

CacheParameterGroup

Specifica un gruppo di parametri che corrisponde alla versione motore. Se stai eseguendo Redis 3.2.4 o versioni successive, specifica il gruppo di parametri `default.redis3.2` o un gruppo di parametri derivato da `default.redis3.2` per creare un gruppo di replica Redis (modalità cluster disabilitata). Per ulteriori informazioni, consulta [Parametri specifici di Redis](#).

--network-type

`ipv4`, `ipv` o `dual-stack`. Se scegli `dual-stack`, devi impostare il parametro `--IpDiscovery` su `ipv4` o `ipv6`.

Motore

`redis`

EngineVersion

`6.0`

I nomi dei nodi verranno derivati dal nome del gruppo di replica aggiungendo -00#. Ad esempio, utilizzando il nome del gruppo di replica myRep1Group, il nome del primario sarà myRep1Group-001 e le repliche di lettura da myRep1Group-002 a myRep1Group-006.

Se desideri abilitare la crittografia dei dati in transito o dei dati inattivi su questo gruppo di replica, aggiungi uno o entrambi i parametri TransitEncryptionEnabled=true o AtRestEncryptionEnabled=true e soddisfa le seguenti condizioni.

- Il gruppo di replica deve eseguire Redis versione 3.2.6 o 4.0.10.
- Il gruppo di replica deve essere creato in un Amazon VPC.
- Devi anche includere il parametro CacheSubnetGroup.
- Deve anche includere il parametro AuthToken con il valore stringa specificato dal cliente per il token AUTH (password) necessario per eseguire operazioni su questo gruppo di replica.

La seguente operazione consente di creare il gruppo di replica Redis (modalità cluster disabilitata) myRep1Group con tre nodi, un nodo primario e due nodi di replica.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateReplicationGroup  
&CacheNodeType=cache.m4.large  
&CacheParameterGroup=default.redis6.x  
&Engine=redis  
&EngineVersion=6.0  
&NumCacheClusters=3  
&ReplicationGroupDescription=test%20group  
&ReplicationGroupId=myRep1Group  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Per informazioni aggiuntive e i parametri da utilizzare, consulta l'argomento [API ElastiCache CreateReplicationGroup](#).

Creazione da zero di un gruppo di replica in Redis (modalità cluster abilitata)

Puoi creare un cluster Redis (modalità cluster abilitata) (API/CLI: gruppo di replica) utilizzando la console ElastiCache, lo AWS CLI o l'API ElastiCache. Un gruppo di replica Redis (modalità cluster abilitata) dispone di un numero di partizioni compreso tra 1 e 500 (API/CLI: gruppi di nodi), un nodo primario in ogni partizione e un massimo di 5 repliche di lettura in ogni partizione. Puoi creare un cluster con un numero più alto di partizioni e un numero più basso di repliche per un totale di 90 nodi per cluster. Questa configurazione del cluster può andare da 90 partizioni e 0 repliche a 15 partizioni e 5 repliche che è il numero massimo consentito di repliche.

Il limite del nodo o della partizione può essere aumentato fino a un massimo di 500 per cluster se la versione del motore Redis è 5.0.6 o superiore. Ad esempio, è possibile scegliere di configurare un cluster a 500 nodi che varia tra 83 partizioni (un primario e 5 repliche per partizione) e 500 partizioni (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. Per ulteriori informazioni, consulta [Creazione di un gruppo di sottoreti](#).

Per le versioni inferiori alla 5.0.6, il limite è 250 per cluster.

Per richiedere un aumento dei limiti, consulta [AWS Limiti del servizio](#) e seleziona il tipo di limite Nodi per cluster per tipo di istanza.

Creazione di un cluster in Redis (modalità cluster abilitata)

- [Creazione di un cluster Redis \(modalità cluster abilitata\) \(console\)](#).
- [Creazione da zero di un gruppo di replica Redis \(modalità cluster abilitata\) \(AWS CLI\)](#)
- [Creazione di un gruppo di replica in Redis \(modalità cluster abilitata\) da zero \(API ElastiCache\)](#)

Creazione di un cluster Redis (modalità cluster abilitata) (console).

Per creare un cluster Redis (modalità cluster abilitata), consulta [Creazione di un cluster Redis \(modalità cluster abilitata\) \(console\)](#). Assicurati di abilitare la modalità cluster, Cluster Mode enabled (Modalità cluster abilitata (scalabilità orizzontale)), e specifica almeno due partizioni e un nodo di replica in ciascuno di essi.

Creazione da zero di un gruppo di replica Redis (modalità cluster abilitata) (AWS CLI)

La procedura seguente crea un gruppo di replica Redis (modalità cluster abilitata) tramite lo AWS CLI

Quando crei un gruppo di replica Redis (modalità cluster abilitata) da zero, crei il gruppo di replica e tutti i relativi nodi vengono creati con una singola chiamata al comando di AWS CLI `create-replication-group`. Includi i seguenti parametri.

`--replication-group-id`

Il nome del gruppo di replica che stai creando.

I vincoli di denominazione dei gruppi di replica Redis (modalità cluster abilitata) 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.

`--replication-group-description`

Descrizione del gruppo di replica.

`--cache-node-type`

Il tipo di nodo per ogni nodo nel gruppo di replica.

ElastiCache supporta i seguenti tipi di nodi. In generale, le generazioni attuali offrono più memoria e più potenza di elaborazione a un costo inferiore rispetto ai prodotti omologhi della generazione precedente.

Per ulteriori informazioni sui dettagli sulle prestazioni per ogni tipo di nodo, consulta [Tipi di istanza di Amazon EC2](#).

`--data-tiering-abilitato`

Impostare questo parametro se si utilizza un tipo di nodo `r6gd`. Se non si desidera il tiering di dati, impostare `--no-data-tiering-enabled`. Per ulteriori informazioni, consulta [Tiering di dati](#).

`--cache-parameter-group`

Specifica il gruppo di parametri `default.redis6.x.cluster.on` un gruppo di parametri derivato da `default.redis6.x.cluster.on` per creare un gruppo di replica Redis (modalità cluster abilitata). Per ulteriori informazioni, consulta [Modifiche dei parametri di Redis 6.x](#).

`--engine`


redis

`--engine-version`

3.2.4

`--num-node-groups`

Il numero di gruppi di nodi in questo gruppo di replica. I valori validi sono compresi tra 1 e 500.

 Note

Il limite dei nodi/partizioni può essere aumentato fino a un massimo di 500 per ogni cluster. Per richiedere un aumento dei limiti, consulta [AWS Limite del servizio](#) e seleziona il tipo di limite “nodi per cluster per tipo di istanza”.

`--replicas-per-node-group`

Il numero di nodi di replica in ogni gruppo di nodi. I valori validi sono compresi tra 0 e 5.

`--network-type`

`ipv4`, `ipv6` o `dual-stack`. Se scegli `dual-stack`, devi impostare il parametro `--IpDiscovery` su `ipv4` o `ipv6`.

Se desideri abilitare la crittografia dei dati in transito o dei dati inattivi su questo gruppo di replica, aggiungi uno o entrambi i parametri `--transit-encryption-enabled` o `--at-rest-encryption-enabled` e soddisfa le seguenti condizioni.

- Il gruppo di replica deve eseguire Redis versione 3.2.6 o 4.0.10.
- Il gruppo di replica deve essere creato in un Amazon VPC.
- Devi anche includere il parametro `--cache-subnet-group`.
- Deve anche includere il parametro `--auth-token` con il valore stringa specificato dal cliente per il token AUTH (password) necessario per eseguire operazioni su questo gruppo di replica.

La seguente operazione consente di creare il gruppo di replica di Redis (modalità cluster abilitata) `sample-repl-group` con tre nodi di gruppi/partizioni (`--num-node-groups`), ciascuno con tre nodi, un nodo primario e due repliche di lettura (`--replicas-per-node-group`).

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id sample-repl-group \  
  --replication-group-description "Demo cluster with replicas" \  
  --num-node-groups 3 \  
  --replicas-per-node-group 2 \  
  --cache-node-type cache.m4.large \  
  --engine redis \  
  --security-group-ids SECURITY_GROUP_ID \  
  --cache-subnet-group-name SUBNET_GROUP_NAME>
```

Per Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --replication-group-description "Demo cluster with replicas" ^  
  --num-node-groups 3 ^  
  --replicas-per-node-group 2 ^  
  --cache-node-type cache.m4.large ^  
  --engine redis ^  
  --security-group-ids SECURITY_GROUP_ID ^  
  --cache-subnet-group-name SUBNET_GROUP_NAME>
```

Il comando precedente genera il seguente output.

```
{  
  "ReplicationGroup": {  
    "Status": "creating",  
    "Description": "Demo cluster with replicas",  
    "ReplicationGroupId": "sample-repl-group",  
    "SnapshotRetentionLimit": 0,  
    "AutomaticFailover": "enabled",  
    "SnapshotWindow": "05:30-06:30",  
    "MemberClusters": [  
      "sample-repl-group-0001-001",  
      "sample-repl-group-0001-002",  
      "sample-repl-group-0001-003",  
      "sample-repl-group-0002-001",  
      "sample-repl-group-0002-002",  
      "sample-repl-group-0002-003",  
      "sample-repl-group-0003-001",
```

```
        "sample-repl-group-0003-002",
        "sample-repl-group-0003-003"
    ],
    "PendingModifiedValues": {}
}
}
```

Quando crei un gruppo di replica Redis (modalità cluster abilitata) da zero, sei in grado di configurare ogni partizione nel cluster utilizzando il parametro `--node-group-configuration` come mostrato nell'esempio seguente che configura due gruppi di nodi (Console: shard). La prima partizione dispone di due nodi, un nodo primario e una replica di lettura. La seconda partizione dispone di tre nodi, un nodo primario e due repliche di lettura.

`--node-group-configuration`

La configurazione per ogni gruppo di nodi. Il parametro `--node-group-configuration` è costituito dai seguenti campi:

- **PrimaryAvailabilityZone** - La zona di disponibilità in cui si trova il nodo primario di questo gruppo di nodi. Se si omette questo parametro, ElastiCache sceglie la zona di disponibilità del nodo primario.

Esempio: `us-west-2a`.

- **ReplicaAvailabilityZones** - Un elenco separato da virgole di zone di disponibilità in cui si trovano le repliche di lettura. Il numero di zone di disponibilità in questo elenco deve corrispondere al valore di **ReplicaCount**. Se si omette il parametro, ElastiCache sceglie le zone di disponibilità per i nodi di replica.

Esempio: `"us-west-2a,us-west-2b,us-west-2c"`

- **ReplicaCount** : Il numero di nodi di replica in questo gruppo di nodi.
- **Slots** - Una stringa che specifica il keyspace per il gruppo di nodi. Il formato della stringa è `startKey-endKey`. Se si omette questo parametro, ElastiCache alloca le chiavi in modo uniforme tra i gruppi di nodi.

Esempio: `"0-4999"`

La seguente operazione consente di creare il gruppo di replica Redis (modalità cluster abilitata) `new-group` con due gruppi di nodi/partizioni (`--num-node-groups`). A differenza dell'esempio precedente, ogni gruppo di nodi è configurato in modo diverso rispetto all'altro gruppo di nodi (`--node-group-configuration`).

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group \
  --replication-group-id new-group \
  --replication-group-description "Sharded replication group" \
  --engine redis \
  --snapshot-retention-limit 8 \
  --cache-node-type cache.m4.medium \
  --num-node-groups 2 \
  --node-group-configuration \
    "ReplicaCount=1,Slots=0-8999,PrimaryAvailabilityZone='us-east-1c',ReplicaAvailabilityZones='us-east-1b'" \
    "ReplicaCount=2,Slots=9000-16383,PrimaryAvailabilityZone='us-east-1a',ReplicaAvailabilityZones='us-east-1a', 'us-east-1c'"
```

Per Windows:

```
aws elasticache create-replication-group ^
  --replication-group-id new-group ^
  --replication-group-description "Sharded replication group" ^
  --engine redis ^
  --snapshot-retention-limit 8 ^
  --cache-node-type cache.m4.medium ^
  --num-node-groups 2 ^
  --node-group-configuration \
    "ReplicaCount=1,Slots=0-8999,PrimaryAvailabilityZone='us-east-1c',ReplicaAvailabilityZones='us-east-1b'" \
    "ReplicaCount=2,Slots=9000-16383,PrimaryAvailabilityZone='us-east-1a',ReplicaAvailabilityZones='us-east-1a', 'us-east-1c'"
```

L'operazione precedente genera il seguente output.

```
{
  "ReplicationGroup": {
    "Status": "creating",
    "Description": "Sharded replication group",
```

```
    "ReplicationGroupId": "rc-rg",
    "SnapshotRetentionLimit": 8,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "10:00-11:00",
    "MemberClusters": [
      "rc-rg-0001-001",
      "rc-rg-0001-002",
      "rc-rg-0002-001",
      "rc-rg-0002-002",
      "rc-rg-0002-003"
    ],
    "PendingModifiedValues": {}
  }
}
```

Per informazioni aggiuntive e i parametri che potrebbe essere necessario utilizzare, consulta l'argomento di AWS CLI [create-replication-group](#).

Creazione di un gruppo di replica in Redis (modalità cluster abilitata) da zero (API ElastiCache)

La seguente procedura consente di creare un gruppo di replica Redis (modalità cluster abilitata) utilizzando l'API ElastiCache.

Quando crei un gruppo di replica Redis (modalità cluster abilitata) da zero, crei il gruppo di replica e tutti i relativi nodi vengono creati con una singola chiamata all'operazione `CreateReplicationGroup` dell'API ElastiCache. Includi i seguenti parametri.

ReplicationGroupId

Il nome del gruppo di replica che stai creando.

I vincoli di denominazione dei gruppi di replica Redis (modalità cluster abilitata) 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.

ReplicationGroupDescription

Descrizione del gruppo di replica.

NumNodeGroups

Il numero di gruppi di nodi che deve essere creato con questo gruppo di replica. I valori validi sono compresi tra 1 e 500.

ReplicasPerNodeGroup

Il numero di nodi di replica in ogni gruppo di nodi. I valori validi sono compresi tra 1 e 5.

NodeGroupConfiguration

La configurazione per ogni gruppo di nodi. Il parametro `NodeGroupConfiguration` è costituito dai seguenti campi:

- `PrimaryAvailabilityZone` - La zona di disponibilità in cui si trova il nodo primario di questo gruppo di nodi. Se si omette questo parametro, ElastiCache sceglie la zona di disponibilità del nodo primario.

Esempio: `us-west-2a`.

- `ReplicaAvailabilityZones` - Un elenco di zone di disponibilità in cui si trovano le repliche di lettura. Il numero di zone di disponibilità in questo elenco deve corrispondere al valore di `ReplicaCount`. Se si omette il parametro, ElastiCache sceglie le zone di disponibilità per i nodi di replica.
- `ReplicaCount` - Il numero di nodi di replica in questo gruppo di nodi.
- `Slots` - Una stringa che specifica il keyspace per il gruppo di nodi. Il formato della stringa è `startKey-endKey`. Se si omette questo parametro, ElastiCache alloca le chiavi in modo uniforme tra i gruppi di nodi.

Esempio: `"0-4999"`

CacheNodeType

Il tipo di nodo per ogni nodo nel gruppo di replica.

ElastiCache supporta i seguenti tipi di nodi. In generale, le generazioni attuali offrono più memoria e più potenza di elaborazione a un costo inferiore rispetto ai prodotti omologhi della generazione precedente.

Per ulteriori informazioni sui dettagli sulle prestazioni per ogni tipo di nodo, consulta [Tipi di istanza di Amazon EC2](#).

--data-tiering-abilitato

Impostare questo parametro se si utilizza un tipo di nodo `r6gd`. Se non si desidera il tiering di dati, impostare `--no-data-tiering-enabled`. Per ulteriori informazioni, consulta [Tiering di dati](#).

CacheParameterGroup

Specifica il gruppo di parametri `default.redis6.x.cluster.on` o un gruppo di parametri derivato da `default.redis6.x.cluster.on` per creare un gruppo di replica Redis (modalità cluster abilitata). Per ulteriori informazioni, consulta [Modifiche dei parametri di Redis 6.x](#).

--network-type

`ipv4`, `ipv` o `dual-stack`. Se scegli `dual-stack`, devi impostare il parametro `--IpDiscovery` su `ipv4` o `ipv6`.

Motore

`redis`

EngineVersion

`6.0`

Se desideri abilitare la crittografia dei dati in transito o dei dati inattivi su questo gruppo di replica, aggiungi uno o entrambi i parametri `TransitEncryptionEnabled=true` o `AtRestEncryptionEnabled=true` e soddisfa le seguenti condizioni.

- Il gruppo di replica deve eseguire Redis versione 3.2.6 o 4.0.10.
- Il gruppo di replica deve essere creato in un Amazon VPC.
- Devi anche includere il parametro `CacheSubnetGroup`.
- Deve anche includere il parametro `AuthToken` con il valore stringa specificato dal cliente per il token AUTH (password) necessario per eseguire operazioni su questo gruppo di replica.

Le interruzioni di riga vengono aggiunte per facilitare la lettura.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateReplicationGroup  
&CacheNodeType=cache.m4.large  
&CacheParameterGroup=default.redis6.xcluster.on
```

```
&Engine=redis
&EngineVersion=6.0
&NumNodeGroups=3
&ReplicasPerNodeGroup=2
&ReplicationGroupDescription=test%20group
&ReplicationGroupId=myReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Per informazioni aggiuntive e i parametri da utilizzare, consulta l'argomento API ElastiCache [CreateReplicationGroup](#).

Visualizzazione dei dettagli del gruppo di replica

A volte potrebbe essere necessario visualizzare i dettagli di un gruppo di replica. È possibile utilizzare la console ElastiCache, lo AWS CLI per ElastiCache o l'API ElastiCache. Il processo di console è diverso per Redis (modalità cluster disabilitata) e Redis (modalità cluster abilitata).

Visualizzazione dei dettagli del gruppo di replica

- [Visualizzazione dei dettagli per un Redis \(modalità cluster disabilitata\) con repliche](#)
 - [Visualizzazione dei dettagli per un gruppo di replica Redis \(modalità cluster disabilitata\) \(console\)](#)
 - [Visualizzazione dei dettagli per un gruppo di replica Redis \(modalità cluster disabilitata\) \(AWS CLI\)](#)
 - [Visualizzazione dei dettagli per un gruppo di replica Redis \(modalità cluster disabilitata\) \(API ElastiCache\)](#)
- [Visualizzazione dei dettagli di un gruppo di replica: Redis \(modalità cluster abilitata\)](#)
 - [Visualizzazione dei dettagli di un cluster Redis \(modalità cluster abilitata\) \(console\)](#)
 - [Visualizzazione dei dettagli di un cluster Redis \(modalità cluster abilitata\) \(AWS CLI\)](#)
 - [Visualizzazione dei dettagli di un cluster Redis \(modalità cluster abilitata\) \(API ElastiCache\)](#)
- [Visualizzazione dei dettagli di un gruppo di replica \(AWS CLI\)](#)
- [Visualizzazione dei dettagli del gruppo di replica \(API ElastiCache\)](#)

Visualizzazione dei dettagli per un Redis (modalità cluster disabilitata) con repliche

È possibile visualizzare i dettagli di un cluster Redis (modalità cluster disabilitata) con repliche (API/CLI: gruppo di replica) utilizzando la console ElastiCache, la AWS CLI per ElastiCache o l'API ElastiCache.

Per visualizzare i dettagli di un cluster Redis (modalità cluster disabilitata)

- [Visualizzazione dei dettagli per un gruppo di replica Redis \(modalità cluster disabilitata\) \(console\)](#)
- [Visualizzazione dei dettagli per un gruppo di replica Redis \(modalità cluster disabilitata\) \(AWS CLI\)](#)
- [Visualizzazione dei dettagli per un gruppo di replica Redis \(modalità cluster disabilitata\) \(API ElastiCache\)](#)

Visualizzazione dei dettagli per un gruppo di replica Redis (modalità cluster disabilitata) (console)

Per visualizzare i dettagli di un cluster Redis (modalità cluster disabilitata) con repliche utilizzando la console ElastiCache, consulta l'argomento [Visualizzazione dei dettagli di un cluster Redis \(modalità cluster disabilitata\) \(console\)](#).

Visualizzazione dei dettagli per un gruppo di replica Redis (modalità cluster disabilitata) (AWS CLI)

Per un AWS CLI in cui vengono visualizzati i dettagli di un gruppo di replica Redis (modalità cluster disabilitata), consulta [Visualizzazione dei dettagli di un gruppo di replica \(AWS CLI\)](#).

Visualizzazione dei dettagli per un gruppo di replica Redis (modalità cluster disabilitata) (API ElastiCache)

Per un esempio di API ElastiCache che visualizza i dettagli di un gruppo di replica Redis (modalità cluster disabilitata), consulta [Visualizzazione dei dettagli del gruppo di replica \(API ElastiCache\)](#).

Visualizzazione dei dettagli di un gruppo di replica: Redis (modalità cluster abilitata)

Visualizzazione dei dettagli di un cluster Redis (modalità cluster abilitata) (console)

Per visualizzare i dettagli di un cluster Redis (modalità cluster abilitata) utilizzando la console ElastiCache, consulta [Visualizzazione dei dettagli di un cluster Redis \(modalità cluster abilitata\) \(console\)](#)

Visualizzazione dei dettagli di un cluster Redis (modalità cluster abilitata) (AWS CLI)

Per un esempio di CLI ElastiCache che visualizza i dettagli di un gruppo di replica Redis (modalità cluster abilitata), consulta [Visualizzazione dei dettagli di un gruppo di replica \(AWS CLI\)](#).

Visualizzazione dei dettagli di un cluster Redis (modalità cluster abilitata) (API ElastiCache)

Per un esempio di API ElastiCache che visualizza i dettagli di un gruppo di replica Redis (modalità cluster abilitata), consulta [Visualizzazione dei dettagli del gruppo di replica \(API ElastiCache\)](#).

Visualizzazione dei dettagli di un gruppo di replica (AWS CLI)

Puoi visualizzare i dettagli per un gruppo di repliche utilizzando il comando `describe-replication-groups` dell'AWS CLI. Utilizza i seguenti parametri opzionali per perfezionare l'elenco. Se si omettono i parametri, vengono restituiti i dettagli per un massimo di 100 gruppi di replica.

Parametri opzionali

- `--replication-group-id` - Utilizza questo parametro per elencare i dettagli di un gruppo di replica specifico. Se il gruppo di replica specificato dispone di più gruppi di nodi, i risultati vengono restituiti raggruppati per gruppo di nodi.
- `--max-items` - Utilizza questo parametro per limitare il numero di gruppi di replica elencati. Il valore di `--max-items` non può essere minore di 20 o maggiore di 100.

Example

Il codice seguente consente di elencare i dettagli per un massimo di 100 gruppi di replica.

```
aws elasticache describe-replication-groups
```

Il codice seguente consente di elencare i dettagli per `sample-repl-group`.

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

Il codice seguente consente di elencare i dettagli per `sample-repl-group`.

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

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

```
aws elasticache describe-replication-groups --max-items 25
```

L'aspetto dell'output di questa operazione è simile al seguente (in formato JSON).

```
{
  "ReplicationGroups": [
    {
      "Status": "available",
      "Description": "test",
      "NodeGroups": [
        {
          "Status": "available",
          "NodeGroupMembers": [
            {
              "CurrentRole": "primary",
              "PreferredAvailabilityZone": "us-west-2a",
              "CacheNodeId": "0001",
              "ReadEndpoint": {
                "Port": 6379,
                "Address": "rg-name-001.1abc4d.0001.usw2.cache.amazonaws.com"
              },
              "CacheClusterId": "rg-name-001"
            },
            {
              "CurrentRole": "replica",
              "PreferredAvailabilityZone": "us-west-2b",
              "CacheNodeId": "0001",
              "ReadEndpoint": {
                "Port": 6379,
                "Address": "rg-name-002.1abc4d.0001.usw2.cache.amazonaws.com"
              },
              "CacheClusterId": "rg-name-002"
            },
            {
              "CurrentRole": "replica",
              "PreferredAvailabilityZone": "us-west-2c",
              "CacheNodeId": "0001",
              "ReadEndpoint": {
                "Port": 6379,
                "Address": "rg-name-003.1abc4d.0001.usw2.cache.amazonaws.com"
              },
              "CacheClusterId": "rg-name-003"
            }
          ],
          "NodeGroupId": "0001",
          "PrimaryEndpoint": {
```

```
        "Port": 6379,
        "Address": "rg-name.1abc4d.ng.0001.usw2.cache.amazonaws.com"
    }
  ],
  "ReplicationGroupId": "rg-name",
  "AutomaticFailover": "enabled",
  "SnapshottingClusterId": "rg-name-002",
  "MemberClusters": [
    "rg-name-001",
    "rg-name-002",
    "rg-name-003"
  ],
  "PendingModifiedValues": {}
},
{
  ... some output omitted for brevity
}
]
}
```

Per ulteriori informazioni, consulta l'argomento AWS CLI per ElastiCache [describe-replication-groups](#).

Visualizzazione dei dettagli del gruppo di replica (API ElastiCache)

Puoi visualizzare i dettagli per un gruppo di replica utilizzando l'operazione `DescribeReplicationGroups` di AWS CLI. Utilizza i seguenti parametri opzionali per perfezionare l'elenco. Se si omettono i parametri, vengono restituiti i dettagli per un massimo di 100 gruppi di replica.

Parametri opzionali

- `ReplicationGroupId` - Utilizza questo parametro per elencare i dettagli di un gruppo di replica specifico. Se il gruppo di replica specificato dispone di più gruppi di nodi, i risultati vengono restituiti raggruppati per gruppo di nodi.
- `MaxRecords` - Utilizza questo parametro per limitare il numero di gruppi di replica elencati. Il valore di `MaxRecords` non può essere minore di 20 o maggiore di 100. Il valore di default è 100.

Example

Il codice seguente consente di elencare i dettagli per un massimo di 100 gruppi di replica.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroups  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Il codice seguente consente di elencare i dettagli per myReplGroup.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroups  
&ReplicationGroupId=myReplGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

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

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroups  
&MaxRecords=25  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta l'argomento di riferimento dell'API ElastiCache [DescribeReplicationGroups](#).

Individuazione di endpoint del gruppo di replica

Un'applicazione può connettersi a qualsiasi nodo in un gruppo di replica, purché disponga dell'endpoint DNS e del numero di porta per tale nodo. A seconda che si stia eseguendo un gruppo di replica Redis (modalità cluster disabilitata) o Redis (modalità cluster abilitata), sarai interessato a endpoint diversi.

Redis (modalità cluster disattivata)

I cluster Redis (modalità cluster disabilitata) con repliche contengono tre tipi di endpoint: l'endpoint primario, l'endpoint di lettura e gli endpoint dei nodi. L'endpoint primario è un nome DNS che viene sempre risolto nel nodo primario nel cluster. L'endpoint primario è immune a modifiche al cluster, ad esempio la promozione di una replica di lettura al ruolo primario. Per attività di scrittura, è preferibile collegare le applicazioni all'endpoint primario.

Un endpoint di lettura dividerà in modo uniforme le connessioni all'endpoint tra tutte le repliche di lettura in un cluster ElastiCache for Redis. Ulteriori fattori come, ad esempio, quando l'applicazione crea le connessioni o il modo in cui l'applicazione riutilizza le connessioni determineranno la distribuzione del traffico. Gli endpoint di lettura mantengono le modifiche ai cluster in tempo reale mano a mano che le repliche vengono aggiunte o rimosse. Puoi posizionare le repliche di lettura multiple del cluster ElastiCache for Redis in diverse zone di disponibilità (AZ) AWS per garantire un'elevata disponibilità degli endpoint di lettura.

Note

Un endpoint di lettura non è un servizio di load balancer. Si tratta di un record DNS che si risolverà in un indirizzo IP di uno dei nodi di replica in modo round robin.

Per attività di lettura, le applicazioni possono inoltre connettersi a qualsiasi nodo nel cluster. A differenza dell'endpoint primario, gli endpoint dei nodi vengono risolti in endpoint specifici. Se apporti una modifica al cluster, ad esempio aggiungendo o eliminando una replica, devi aggiornare gli endpoint dei nodi nell'applicazione.

Redis (Modalità cluster abilitata)

Cluster Redis (modalità cluster abilitata) con repliche, perché dispongono di più partizioni (API/CLI: gruppi di nodi), il che significa che dispongono anche di più nodi primari, una struttura di endpoint diversa dai cluster Redis (modalità cluster disabilitata). Redis (modalità cluster abilitata) ha un

endpoint di configurazione che «conosce» tutti gli endpoint primari e nodi nel cluster. L'applicazione si connette all'endpoint di configurazione. Ogni volta che l'applicazione esegue operazioni di scrittura e lettura dall'endpoint di configurazione del cluster, Redis, dietro le quinte, determina a quale partizione appartiene la chiave e quale endpoint della partizione da utilizzare. Tutto è trasparente all'applicazione.

È possibile visualizzare i dettagli di un cluster utilizzando la console ElastiCache, la AWS CLI o l'API ElastiCache.

Individuazione di endpoint del gruppo di replica

Per trovare gli endpoint per il gruppo di replica, consulta uno dei seguenti argomenti:

- [Individuazione degli endpoint di un cluster Redis \(modalità cluster disabilitata\) \(console\)](#)
- [Individuazione degli endpoint di un cluster Redis \(modalità cluster abilitata\) \(console\)](#)
- [Individuazione degli endpoint per gruppi di replica \(AWS CLI\)](#)
- [Individuazione di endpoint dei gruppi di replica \(API ElastiCache\)](#)

Modifica di un gruppo di replica

Vincoli importanti

- Attualmente, ElastiCache supporta modifiche limitate di un gruppo di replica Redis (modalità cluster abilitata), ad esempio la modifica della versione motore, utilizzando l'operazione API `ModifyReplicationGroup` (CLI: `modify-replication-group`). Puoi modificare il numero di shard (gruppi di nodi) in un cluster Redis (modalità cluster abilitata) con l'operazione API [ModifyReplicationGroupShardConfiguration](#) (CLI: `modify-replication-group-shard-configuration`). Per ulteriori informazioni, consulta [Ridimensionamento dei cluster in Redis \(modalità cluster attivata\)](#).

Altre modifiche a un cluster Redis (modalità cluster abilitata) richiedono la creazione di un cluster in cui sono incorporate le modifiche.

- Puoi aggiornare cluster e gruppi di replica Redis (modalità cluster disabilitata) e Redis (modalità cluster abilitata) alle nuove versioni del motore. Tuttavia, non puoi eseguire il downgrade alle versioni precedenti del motore se non eliminando il cluster esistente o il gruppo di replica e creandolo di nuovo. Per ulteriori informazioni, consulta [Versioni e aggiornamenti del motore](#).
- Puoi aggiornare un cluster ElastiCache per Redis esistente che utilizza la modalità cluster disabilitata per utilizzare la modalità cluster abilitata, mediante la console, l'API [ModifyReplication](#) o il comando della CLI `modify-replication-group`, come mostrato nell'esempio seguente. Oppure, puoi seguire la procedura descritta in [Modifica della modalità cluster](#).

È possibile modificare i dettagli di un cluster Redis (modalità cluster disabilitata) utilizzando la console ElastiCache, la AWS CLI o l'API ElastiCache. Attualmente, ElastiCache supporta un numero limitato di modifiche in un gruppo di replica Redis (modalità cluster abilitata). Altre modifiche richiedono la creazione di un backup del gruppo di replica corrente e quindi l'utilizzo di tale backup per inizializzare un nuovo gruppo di replica Redis (modalità cluster abilitata).

Argomenti

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

Utilizzo di AWS Management Console

Per modificare un cluster Redis (modalità cluster disabilitata), vedi [Modifica di un cluster ElastiCache](#).

Utilizzo di AWS CLI

Di seguito sono riportati esempi AWS CLI del comando `modify-replication-group`. Puoi utilizzare lo stesso comando per apportare altre modifiche a un gruppo di replica.

Abilita Multi-AZ su un gruppo di replica Redis esistente:

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --multi-az-enabled = true
```

Per Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --multi-az-enabled
```

Modifica della modalità cluster da disabilitata ad abilitata:

Per modificare la modalità cluster da disabilitata ad abilitata, è necessario innanzitutto impostare la modalità cluster su compatibile. La modalità compatibile consente ai client Redis di connettersi utilizzando la modalità cluster abilitata e la modalità cluster disabilitata. Dopo aver migrato tutti i client Redis per utilizzare la modalità cluster abilitata, è possibile completare la configurazione della modalità cluster e impostare la modalità cluster su abilitata.

Per Linux, macOS o Unix:

Imposta la modalità cluster su compatibile.

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --cache-parameter-group-name myParameterGroupName \  
  --cluster-mode compatible
```

Imposta la modalità cluster su abilitata.

```
aws elasticache modify-replication-group \  
  --replication-group-id myReplGroup \  
  --cluster-mode enabled
```

Per Windows:

Imposta la modalità cluster su compatibile.

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --cache-parameter-group-name myParameterGroupName ^  
  --cluster-mode compatible
```

Imposta la modalità cluster su abilitata.

```
aws elasticache modify-replication-group ^  
  --replication-group-id myReplGroup ^  
  --cluster-mode enabled
```

Per ulteriori informazioni sul comando AWS CLI `modify-replication-group`, consultare [modify-replication-group](#) o [Modifica della modalità cluster](#) nella Guida per l'utente di ElastiCache per Redis.

Utilizzo dell'API ElastiCache

La seguente operazione API ElastiCache abilita Multi-AZ su un gruppo di replica Redis esistente. Puoi utilizzare la stessa operazione per apportare altre modifiche a un gruppo di replica.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyReplicationGroup  
  &AutomaticFailoverEnabled=true  
  &Mutli-AZEnabled=true  
  &ReplicationGroupId=myReplGroup  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &Version=2014-12-01  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>
```

```
&X-Amz-Signature=<signature>
```

Per ulteriori informazioni sull'operazione dell'API ElastiCache `ModifyReplicationGroup`, consulta [ModifyReplicationGroup](#).

Eliminazione di un gruppo di replica

Se non hai più bisogno dei cluster con repliche (chiamati gruppi di replica nell'API/CLI), puoi eliminarli. Quando elimini un gruppo di replica, ElastiCache elimina tutti i nodi in tale gruppo.

Dopo che è stata avviata, questa operazione non può essere interrotta o annullata.

Warning

Quando elimini un cluster ElastiCache, vengono conservati i tuoi snapshot manuali. Hai anche la possibilità di creare uno snapshot finale prima che il cluster venga eliminato. Gli snapshot di cache automatici non vengono conservati.

Eliminazione di un gruppo di replica (console)

Per eliminare un cluster che dispone di repliche, consulta [Eliminazione di un cluster](#).

Eliminazione di un gruppo di replica (AWS CLI)

Utilizza il comando [delete-replication-group](#) per eliminare un gruppo di replica.

```
aws elasticache delete-replication-group --replication-group-id my-repgroup
```

Ti viene chiesto di confermare la tua decisione. Immetti y (sì) per avviare l'operazione immediatamente. Una volta avviato, il processo è irreversibile.

```
After you begin deleting this replication group, all of its nodes will be deleted as well.
```

```
Are you sure you want to delete this replication group? [Ny]y
```

```
REPLICATIONGROUP my-repgroup My replication group deleting
```

Eliminazione di un gruppo di replica (API ElastiCache)

Chiama [DeleteReplicationGroup](#) con il parametro `ReplicationGroup`.

Example

```
https://elasticache.us-west-2.amazonaws.com/
```

```
?Action=DeleteReplicationGroup
&ReplicationGroupId=my-repgroup
&Version=2014-12-01
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

Note

Se imposti il parametro `RetainPrimaryCluster` su `true`, tutte le repliche di lettura verranno eliminate, ma il cluster primario verrà mantenuto.

Modifica del numero di repliche

Puoi aumentare o diminuire dinamicamente il numero di repliche di lettura nel gruppo di replica Redis utilizzando lo AWS Management Console, la AWS CLI o l'API ElastiCache. Se il gruppo di replica è un gruppo di replica (modalità cluster abilitata) Redis, puoi scegliere le partizioni (gruppi di nodi) per cui aumentare o diminuire il numero di repliche.

Per modificare dinamicamente il numero di repliche nel gruppo di replica Redis, dalla tabella seguente scegliere l'operazione più adatta alla situazione.

Per	Redis (Modalità cluster abilitata)	Redis (modalità cluster disattivata)
Aggiungere repliche	Incremento del numero di repliche in uno shard	Incremento del numero di repliche in uno shard Aggiunta di una replica di lettura per i gruppi di replica Redis (modalità cluster disabilitata)
Eliminare repliche	Riduzione del numero di repliche in uno shard	Riduzione del numero di repliche in uno shard Eliminazione di una replica di lettura, per i gruppi di replica Redis (modalità cluster disabilitata)

Incremento del numero di repliche in uno shard

Puoi aumentare il numero di repliche in uno shard Redis (modalità cluster abilitata) o Redis (modalità cluster disabilitata) fino a un massimo di cinque. A questo scopo, utilizza lo AWS Management Console, la AWS CLI o l'API ElastiCache.

Argomenti

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

Utilizzo di AWS Management Console

La procedura seguente utilizza la console per aumentare il numero di repliche in un gruppo di replica Redis (modalità cluster abilitata).

Per aumentare il numero di repliche in shard Redis

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegliere Redis, quindi scegliere il nome del gruppo di replica cui aggiungere le repliche.
3. Scegliere la casella per ogni shard cui si desidera aggiungere repliche.
4. Scegliere Add replicas (Aggiungi repliche).
5. Completare la pagina Add Replicas to Shards (Aggiungi repliche agli shard):
 - Per New number of replicas/shard (Nuovo numero di repliche/shard), immettere il numero di repliche che devono avere tutti gli shard selezionati. Questo valore deve essere maggiore o uguale di Current Number of Replicas per shard (Numero attuale di repliche per shard) e minore o uguale a cinque. Si consiglia come minimo due repliche.
 - Per Availability Zones (Zone di disponibilità), scegliere No preference (Nessuna preferenza) affinché ElastiCache scelga una zona di disponibilità per ogni nuova replica, oppure Specify Availability Zones (Specifica zone di disponibilità) per scegliere una zona di disponibilità per ogni nuova replica.

Se si sceglie Specify Availability Zones (Specifica zone di disponibilità), per ogni nuova replica specificare una zona di disponibilità utilizzando l'elenco.

6. Scegliere Add (Aggiungi) per aggiungere le repliche o Cancel (Annulla) per annullare l'operazione.

Utilizzo di AWS CLI

Per aumentare il numero di repliche in uno shard Redis, utilizza il comando `increase-replica-count` con i seguenti parametri:

- `--replication-group-id` - Obbligatorio. Identifica il gruppo di replica in cui incrementare il numero di repliche.
- `--apply-immediately` o `--no-apply-immediately` - Obbligatorio. Specifica se aumentare il conteggio di repliche immediatamente (`--apply-immediately`) o alla successiva finestra di manutenzione (`--no-apply-immediately`). Attualmente, `--no-apply-immediately` non è supportato.
- `--new-replica-count` - Opzionale. Specifica il numero di nodi di replica desiderati, fino a un massimo di cinque. Utilizza questo parametro per gruppi di replica per Redis (modalità cluster disabilitata) in cui esiste un solo gruppo di nodi o gruppo o in cui tutti i gruppi di nodi devono avere lo stesso numero di repliche. Se questo valore non è maggiore del numero corrente di repliche nel gruppo di nodi, la chiamata non riesce con un'eccezione.
- `--replica-configuration` - Opzionale. Consente di impostare il numero di repliche e zone di disponibilità per ogni gruppo di nodi in maniera indipendente. Utilizza questo parametro per gruppi Redis (modalità cluster abilitata) in cui desideri configurare ciascun gruppo di nodi in maniera indipendente.

`--replica-configuration` dispone di tre membri opzionali:

- `NodeGroupId` - L'ID di quattro cifre del gruppo di nodi che stai configurando. Per i gruppi di replica Redis (modalità cluster disabilitata), l'ID dello shard è sempre `0001`. Per trovare l'ID di un gruppo di nodi Redis (modalità cluster abilitata), consulta [Ricerca dell'ID di uno shard](#).
- `NewReplicaCount` - Il numero di repliche desiderato in questo gruppo di nodi al termine di questa operazione. Il valore deve essere maggiore del numero corrente di repliche, fino a un massimo di cinque. Se questo valore non è maggiore del numero corrente di repliche nel gruppo di nodi, la chiamata non riesce con un'eccezione.
- `PreferredAvailabilityZones` - Un elenco di stringhe `PreferredAvailabilityZone` che specifica in quali zone di disponibilità si devono trovare i nodi del gruppo di replica. Il numero di valori `PreferredAvailabilityZone` deve essere uguale al valore di `NewReplicaCount` più 1 per tenere conto del nodo primario. Se si omette questo membro di `--replica-`

configuration, ElastiCache for Redis sceglie la zona di disponibilità per ciascuna delle nuove repliche.

⚠ Important

Nella chiamata è necessario includere il parametro `--new-replica-count` o `--replica-configuration`, ma non entrambi.

Example

Nell'esempio seguente il numero di repliche nel gruppo di replica `sample-repl-group` viene incrementato a tre. Al termine dell'esempio, ogni gruppo di nodi contiene tre repliche. Questo numero vale nel caso di un gruppo Redis (modalità cluster disabilitata) con un singolo gruppo di nodi o un gruppo Redis (modalità cluster abilitata) con più gruppi di nodi.

Per Linux, macOS o Unix:

```
aws elasticache increase-replica-count \  
  --replication-group-id sample-repl-group \  
  --new-replica-count 3 \  
  --apply-immediately
```

Per Windows:

```
aws elasticache increase-replica-count ^  
  --replication-group-id sample-repl-group ^  
  --new-replica-count 3 ^  
  --apply-immediately
```

Nell'esempio seguente il numero di repliche nel gruppo di replica `sample-repl-group` viene incrementato al valore specificato per i due gruppi di nodi specificati. Dato che esistono più gruppi di nodi, questo è un gruppo di replica Redis (modalità cluster abilitata). Durante la specifica del `PreferredAvailabilityZones` opzionale, il numero di zone di disponibilità elencate deve essere uguale al valore di `NewReplicaCount` più uno. Questo approccio consente di tenere conto del nodo primario per il gruppo identificato da `NodeGroupId`.

Per Linux, macOS o Unix:

```
aws elasticache increase-replica-count \  
  --replication-group-id sample-repl-group \  
  --replica-configuration \  
    NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c,us-east-1b \  
    NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c,us-east-1c \  
  --apply-immediately
```

Per Windows:

```
aws elasticache increase-replica-count ^  
  --replication-group-id sample-repl-group ^  
  --replica-configuration ^  
    NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c,us-east-1b ^  
    NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c,us-east-1c \  
  --apply-immediately
```

Per ulteriori informazioni sull'aumento del numero di repliche mediante la CLI, consulta [increase-replica-count](#) nella guida di riferimento alla riga di comando Amazon ElastiCache.

Utilizzo dell'API ElastiCache

Per aumentare il numero di repliche in uno shard Redis, utilizza l'operazione `IncreaseReplicaCount` con i seguenti parametri:

- `ReplicationGroupId` - Obbligatorio Identifica il gruppo di replica in cui incrementare il numero di repliche.
- `ApplyImmediately` – Obbligatorio Specifica se aumentare il conteggio di repliche immediatamente (`ApplyImmediately=True`) o alla successiva finestra di manutenzione (`ApplyImmediately=False`). Attualmente, `ApplyImmediately=False` non è supportato.
- `NewReplicaCount` - Opzionale. Specifica il numero di nodi di replica desiderati, fino a un massimo di cinque. Utilizza questo parametro per gruppi di replica Redis (modalità cluster disabilitata) in cui esiste un solo gruppo di nodi o gruppi Redis (modalità cluster abilitata) o in cui tutti i gruppi di nodi devono avere lo stesso numero di repliche. Se questo valore non è maggiore del numero corrente di repliche nel gruppo di nodi, la chiamata non riesce con un'eccezione.

- **ReplicaConfiguration** - Opzionale. Consente di impostare il numero di repliche e zone di disponibilità per ogni gruppo di nodi in maniera indipendente. Utilizza questo parametro per gruppi Redis (modalità cluster abilitata) in cui desideri configurare ciascun gruppo di nodi in maniera indipendente.

ReplicaConfiguration dispone di tre membri opzionali:

- **NodeId** - L'ID di quattro cifre del gruppo di nodi che si sta configurando. Per i gruppi di replica Redis (modalità cluster disabilitata), l'ID del gruppo di nodi (shard) è sempre 0001. Per trovare l'ID di un gruppo di nodi Redis (modalità cluster abilitata), consulta [Ricerca dell'ID di uno shard](#).
- **NewReplicaCount** - Il numero di repliche desiderato in questo gruppo di nodi al termine di questa operazione. Il valore deve essere maggiore del numero corrente di repliche e non superiore a cinque. Se questo valore non è maggiore del numero corrente di repliche nel gruppo di nodi, la chiamata non riesce con un'eccezione.
- **PreferredAvailabilityZones** - Un elenco di stringhe **PreferredAvailabilityZone** che specifica in quali zone di disponibilità si devono trovare i nodi del gruppo di replica. Il numero di valori **PreferredAvailabilityZone** deve essere uguale al valore di **NewReplicaCount** più 1 per tenere conto del nodo primario. Se si omette questo membro di **ReplicaConfiguration**, ElastiCache for Redis sceglie la zona di disponibilità per ciascuna delle nuove repliche.

Important

Nella chiamata è necessario includere il parametro **NewReplicaCount** o **ReplicaConfiguration**, ma non entrambi.

Example

Nell'esempio seguente il numero di repliche nel gruppo di replica `sample-repl-group` viene incrementato a tre. Al termine dell'esempio, ogni gruppo di nodi contiene tre repliche. Questo numero vale nel caso di un gruppo Redis (modalità cluster disabilitata) con un singolo gruppo di nodi o un gruppo Redis (modalità cluster abilitata) con più gruppi di nodi.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=IncreaseReplicaCount  
&ApplyImmediately=True
```

```

&NewReplicaCount=3
&ReplicationGroupId=sample-repl-group
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>

```

Nell'esempio seguente il numero di repliche nel gruppo di replica `sample-repl-group` viene incrementato al valore specificato per i due gruppi di nodi specificati. Dato che esistono più gruppi di nodi, questo è un gruppo di replica Redis (modalità cluster abilitata). Durante la specifica del `PreferredAvailabilityZones` opzionale, il numero di zone di disponibilità elencate deve essere uguale al valore di `NewReplicaCount` più uno. Questo approccio consente di tenere conto del nodo primario per il gruppo identificato da `NodeGroupId`.

```

https://elasticache.us-west-2.amazonaws.com/
  ?Action=IncreaseReplicaCount
  &ApplyImmediately=True
  &ReplicaConfiguration.ConfigureShard.1.NodeGroupId=0001
  &ReplicaConfiguration.ConfigureShard.1.NewReplicaCount=2

  &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

  &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1c

  &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.3=
east-1b
  &ReplicaConfiguration.ConfigureShard.2.NodeGroupId=0003
  &ReplicaConfiguration.ConfigureShard.2.NewReplicaCount=3

  &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

  &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1b

  &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.3=
east-1c

```

```
&ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.4=  
east-1c  
  &ReplicationGroupId=sample-repl-group  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Per ulteriori informazioni sull'aumento del numero di repliche mediante l'API, consulta [IncreaseReplicaCount](#) nel Riferimento API Amazon ElastiCache

Riduzione del numero di repliche in uno shard

Puoi ridurre il numero di repliche in uno shard per Redis (modalità cluster abilitata) o in un gruppo di replica per Redis (modalità cluster disabilitata):

- Per Redis (modalità cluster disabilitata), puoi ridurre il numero di repliche a uno se la funzione Multi-AZ è abilitata e a zero se non è abilitata.
- Per Redis (modalità cluster abilitata), puoi ridurre il numero di repliche a zero. Tuttavia, non puoi eseguire il failover in una replica se il nodo primario non riesce.

Puoi utilizzare la AWS Management Console, la AWS CLI o l'API ElastiCache per ridurre il numero di repliche in un gruppo di nodi (shard) o gruppo di replica.

Argomenti

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

Utilizzo di AWS Management Console

La procedura seguente utilizza la console per ridurre il numero di repliche in un gruppo di replica Redis (modalità cluster abilitata).

Per ridurre il numero di repliche in uno shard Redis

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegliere Redis, quindi scegliere il nome del gruppo di replica da cui si desidera eliminare repliche.
3. Scegliere la casella per ogni shard da cui si desidera rimuovere un nodo di replica.
4. Scegliere Delete replicas (Elimina repliche).
5. Completare la pagina Delete Replicas from Shards (Elimina repliche dagli shard):
 - a. Per New number of replicas/shard (Nuovo numero di repliche/shard), immettere il numero di repliche specificato per gli shard selezionati. Questo numero deve essere maggiore o uguale a 1. Si consiglia almeno due repliche per shard.

- b. Scegliere Delete (Elimina) per eliminare le repliche o Cancel (Annulla) per annullare l'operazione.

Important

- Se non si specificano i nodi di replica da eliminare, ElastiCache for Redis seleziona automaticamente i nodi di replica per l'eliminazione. Così facendo, ElastiCache for Redis tenta di conservare l'architettura Multi-AZ per il gruppo di replica seguito dal mantenimento di repliche con ritardo di replica minimo con il master.
- Non è possibile eliminare il primario o nodi primari in un gruppo di replica. Se si specifica un nodo primario per l'eliminazione, l'operazione non riesce e viene restituito un errore che indica che il nodo primario è stato selezionato per l'eliminazione.


Utilizzo di AWS CLI

Per ridurre il numero di repliche in uno shard Redis, utilizza il comando `decrease-replica-count` con i seguenti parametri:

- `--replication-group-id` : Obbligatorio Identifica il gruppo di replica in cui ridurre il numero di repliche.
- `--apply-immediately` o `--no-apply-immediately` : Obbligatorio. Specifica se ridurre il conteggio di repliche immediatamente (`--apply-immediately`) o alla successiva finestra di manutenzione (`--no-apply-immediately`). Attualmente, `--no-apply-immediately` non è supportato.
- `--new-replica-count` - Opzionale. Specifica il numero di nodi di replica desiderato. Il valore di `--new-replica-count` deve essere un valore valido minore del numero corrente di repliche nei gruppi di nodi. Per valori consentiti minimi, consulta [Riduzione del numero di repliche in uno shard](#). Se il valore di `--new-replica-count` non soddisfa questo requisiti, la chiamata non riesce.
- `--replicas-to-remove` : Opzionale. Contiene un elenco di ID nodo che specificano i nodi di replica da rimuovere.
- `--replica-configuration` - Opzionale. Consente di impostare il numero di repliche e zone di disponibilità per ogni gruppo di nodi in maniera indipendente. Utilizza questo parametro per gruppi Redis (modalità cluster abilitata) in cui desideri configurare ciascun gruppo di nodi in maniera indipendente.

`--replica-configuration` dispone di tre membri opzionali:

- `NodeGroupId` - L'ID di quattro cifre del gruppo di nodi che stai configurando. Per i gruppi di replica Redis (modalità cluster disabilitata), l'ID dello shard è sempre `0001`. Per trovare l'ID di un gruppo di nodi Redis (modalità cluster abilitata), consulta [Ricerca dell'ID di uno shard](#).
- `NewReplicaCount` - Un parametro facoltativo che specifica il numero di nodi di replica desiderati. Il valore di `NewReplicaCount` deve essere un valore valido minore del numero corrente di repliche nei gruppi di nodi. Per valori consentiti minimi, consulta [Riduzione del numero di repliche in uno shard](#). Se il valore di `NewReplicaCount` non soddisfa questo requisiti, la chiamata non riesce.
- `PreferredAvailabilityZones` - Un elenco di stringhe `PreferredAvailabilityZone` che specifica in quali zone di disponibilità si trovano i nodi del gruppo di replica. Il numero di valori `PreferredAvailabilityZone` deve essere uguale al valore di `NewReplicaCount` più 1 per tenere conto del nodo primario. Se si omette questo membro di `ElastiCache`, `--replica-configuration` sceglie la zona di disponibilità per ciascuna delle nuove repliche.

 Important

Devi includere solo uno dei parametri `--new-replica-count`, `--replicas-to-remove` o `--replica-configuration`.

Example

L'esempio seguente utilizza `--new-replica-count` per ridurre a uno il numero di repliche nel gruppo di replica `sample-repl-group`. Al termine dell'esempio, ogni gruppo di nodi contiene una sola replica. Questo numero vale nel caso di un gruppo Redis (modalità cluster disabilitata) con un singolo gruppo di nodi o un gruppo Redis (modalità cluster abilitata) con più gruppi di nodi.

Per Linux, macOS o Unix:

```
aws elasticache decrease-replica-count
  --replication-group-id sample-repl-group \
  --new-replica-count 1 \
  --apply-immediately
```

Per Windows:

```
aws elasticache decrease-replica-count ^
  --replication-group-id sample-repl-group ^
  --new-replica-count 1 ^
  --apply-immediately
```

Nell'esempio seguente il numero di repliche nel gruppo di replica `sample-repl-group` viene ridotto rimuovendo due repliche specificate (`0001` e `0003`) dal gruppo di nodi.

Per Linux, macOS o Unix:

```
aws elasticache decrease-replica-count \
  --replication-group-id sample-repl-group \
  --replicas-to-remove 0001,0003 \
  --apply-immediately
```

Per Windows:

```
aws elasticache decrease-replica-count ^
  --replication-group-id sample-repl-group ^
  --replicas-to-remove 0001,0003 \
  --apply-immediately
```

L'esempio seguente utilizza `--replica-configuration` per ridurre il numero di repliche nel gruppo di replica `sample-repl-group` al valore specificato per i due gruppi di nodi specificati. Dato che esistono più gruppi di nodi, questo è un gruppo di replica Redis (modalità cluster abilitata). Durante la specifica del `PreferredAvailabilityZones` opzionale, il numero di zone di disponibilità elencate deve essere uguale al valore di `NewReplicaCount` più uno. Questo approccio consente di tenere conto del nodo primario per il gruppo identificato da `NodeGroupId`.

Per Linux, macOS o Unix:

```
aws elasticache decrease-replica-count \
  --replication-group-id sample-repl-group \
  --replica-configuration \
    NodeGroupId=0001,NewReplicaCount=1,PreferredAvailabilityZones=us-east-1a,us-  
east-1c \
    NodeGroupId=0003,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-  
east-1b,us-east-1c \
  --apply-immediately
```

Per Windows:

```
aws elasticache decrease-replica-count ^
  --replication-group-id sample-repl-group ^
  --replica-configuration ^
    NodeGroupId=0001,NewReplicaCount=2,PreferredAvailabilityZones=us-east-1a,us-east-1c ^
    NodeGroupId=0003,NewReplicaCount=3,PreferredAvailabilityZones=us-east-1a,us-east-1b,us-east-1c \
  --apply-immediately
```

Per ulteriori informazioni sulla riduzione del numero di repliche mediante la CLI, consulta [decrease-replica-count](#) nella guida di riferimento alla riga di comando Amazon ElastiCache

Utilizzo dell'API ElastiCache

Per ridurre il numero di repliche in uno shard Redis, utilizza l'operazione `DecreaseReplicaCount` con i seguenti parametri:

- `ReplicationGroupId` - Obbligatorio Identifica il gruppo di replica in cui ridurre il numero di repliche.
- `ApplyImmediately` - Obbligatorio Specifica se ridurre il conteggio di repliche immediatamente (`ApplyImmediately=True`) o alla successiva finestra di manutenzione (`ApplyImmediately=False`). Attualmente, `ApplyImmediately=False` non è supportato.
- `NewReplicaCount` - Opzionale. Specifica il numero di nodi di replica desiderato. Il valore di `NewReplicaCount` deve essere un valore valido minore del numero corrente di repliche nei gruppi di nodi. Per valori consentiti minimi, consulta [Riduzione del numero di repliche in uno shard](#). Se il valore di `--new-replica-count` non soddisfa questo requisiti, la chiamata non riesce.
- `ReplicasToRemove` : Opzionale. Contiene un elenco di ID nodo che specificano i nodi di replica da rimuovere.
- `ReplicaConfiguration` : Opzionale. Contiene un elenco di gruppi di nodi che consentono di impostare il numero di repliche e zone di disponibilità per ogni gruppo di nodi in maniera indipendente. Utilizza questo parametro per gruppi Redis (modalità cluster abilitata) in cui desideri configurare ciascun gruppo di nodi in maniera indipendente.

`ReplicaConfiguration` dispone di tre membri opzionali:

- `NodeGroupId` - L'ID di quattro cifre del gruppo di nodi che si sta configurando. Per i gruppi di replica Redis (modalità cluster disabilitata), l'ID del gruppo di nodi è sempre `0001`. Per trovare l'ID di un gruppo di nodi Redis (modalità cluster abilitata), consulta [Ricerca dell'ID di uno shard](#).

- **NewReplicaCount** - Il numero di repliche desiderato in questo gruppo di nodi al termine di questa operazione. Il valore deve essere inferiore al numero di repliche correnti fino a un minimo di 1 se la funzione Multi-AZ è abilitata o 0 se non è abilitata failover automatico. Se questo valore non è minore del numero corrente di repliche nel gruppo di nodi, la chiamata non riesce con un'eccezione.
- **PreferredAvailabilityZones** - Un elenco di stringhe PreferredAvailabilityZone che specifica in quali zone di disponibilità si trovano i nodi del gruppo di replica. Il numero di valori PreferredAvailabilityZone deve essere uguale al valore di NewReplicaCount più 1 per tenere conto del nodo primario. Se si omette questo membro di ElastiCache, ReplicaConfiguration sceglie la zona di disponibilità per ciascuna delle nuove repliche.

Important

Devi includere solo uno dei parametri NewReplicaCount, ReplicasToRemove o ReplicaConfiguration.

Example

L'esempio seguente utilizza NewReplicaCount per ridurre a uno il numero di repliche nel gruppo di replica `sample-repl-group`. Al termine dell'esempio, ogni gruppo di nodi contiene una sola replica. Questo numero vale nel caso di un gruppo Redis (modalità cluster disabilitata) con un singolo gruppo di nodi o un gruppo Redis (modalità cluster abilitata) con più gruppi di nodi.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DecreaseReplicaCount  
  &ApplyImmediately=True  
  &NewReplicaCount=1  
  &ReplicationGroupId=sample-repl-group  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Nell'esempio seguente il numero di repliche nel gruppo di replica `sample-repl-group` viene ridotto rimuovendo due repliche specificate (`0001` e `0003`) dal gruppo di nodi.

```

https://elasticache.us-west-2.amazonaws.com/
  ?Action=DecreaseReplicaCount
  &ApplyImmediately=True
  &ReplicasToRemove.ReplicaToRemove.1=0001
  &ReplicasToRemove.ReplicaToRemove.2=0003
  &ReplicationGroupId=sample-repl-group
  &Version=2015-02-02
  &SignatureVersion=4
  &SignatureMethod=HmacSHA256
  &Timestamp=20150202T192317Z
  &X-Amz-Credential=<credential>

```

L'esempio seguente utilizza `ReplicaConfiguration` per ridurre il numero di repliche nel gruppo di replica `sample-repl-group` al valore specificato per i due gruppi di nodi specificati. Dato che esistono più gruppi di nodi, questo è un gruppo di replica Redis (modalità cluster abilitata). Durante la specifica del `PreferredAvailabilityZones` opzionale, il numero di zone di disponibilità elencate deve essere uguale al valore di `NewReplicaCount` più uno. Questo approccio consente di tenere conto del nodo primario per il gruppo identificato da `NodeGroupId`.

```

https://elasticache.us-west-2.amazonaws.com/
  ?Action=DecreaseReplicaCount
  &ApplyImmediately=True
  &ReplicaConfiguration.ConfigureShard.1.NodeGroupId=0001
  &ReplicaConfiguration.ConfigureShard.1.NewReplicaCount=1

  &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

  &ReplicaConfiguration.ConfigureShard.1.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1c
  &ReplicaConfiguration.ConfigureShard.2.NodeGroupId=0003
  &ReplicaConfiguration.ConfigureShard.2.NewReplicaCount=2

  &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.1=
east-1a

  &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.2=
east-1b

  &ReplicaConfiguration.ConfigureShard.2.PreferredAvailabilityZones.PreferredAvailabilityZone.4=
east-1c
  &ReplicationGroupId=sample-repl-group

```

```
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni sulla riduzione del numero di repliche mediante l'API, consulta [DecreaseReplicaCount](#) nella Guida di riferimento dell'API Amazon ElastiCache

Aggiunta di una replica di lettura per i gruppi di replica Redis (modalità cluster disabilitata)

Le informazioni nel seguente argomento sono valide solo per gruppi di replica Redis (modalità cluster disabilitata) .

All'aumentare del traffico di lettura, potrebbe essere necessario distribuire tali letture tra più nodi e ridurre la pressione di lettura su qualsiasi nodo. In questo argomento, viene descritto come aggiungere una replica di lettura a un cluster Redis (modalità cluster disabilitata).

Un gruppo di replica Redis (modalità cluster disattivata) può disporre di un massimo di cinque repliche di lettura. Se tenti di aggiungere una replica di lettura a un gruppo di replica che dispone già di cinque repliche di lettura, l'operazione non riesce.

Per informazioni sull'aggiunta di repliche a un gruppo di replica Redis (modalità cluster abilitata), consulta la documentazione seguente.

- [Ridimensionamento dei cluster in Redis \(modalità cluster attivata\)](#)
- [Incremento del numero di repliche in uno shard](#)

Puoi aggiungere una replica di lettura a un cluster Redis (modalità cluster disabilitata) utilizzando la console ElastiCache, la AWS CLI o l'API ElastiCache.

Argomenti correlati

- [Aggiunta di nodi a un cluster](#)
- [Aggiunta di una replica di lettura a un gruppo di replica \(AWS CLI\)](#)
- [Aggiunta di una replica di lettura a un gruppo di replica utilizzando l'API](#)

Aggiunta di una replica di lettura a un gruppo di replica (AWS CLI)

Per aggiungere una replica di lettura a un gruppo di replica Redis (modalità cluster disabilitata), utilizza il comando AWS CLI di `create-cache-cluster` con il parametro `--replication-group-id` per specificare il gruppo di replica cui aggiungere il cluster (nodo).

L'esempio seguente consente di creare il cluster `my-read-replica` e di aggiungerlo al gruppo di replica `my-replication-group`. I tipi di nodi, i gruppi di parametri, i gruppi di sicurezza, la finestra di manutenzione e altre impostazioni per la replica di lettura sono identiche a quelle di altri nodi in `my-replication-group`.

Per Linux, macOS o Unix:

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id my-read-replica \  
  --replication-group-id my-replication-group
```

Per Windows:

```
aws elasticache create-cache-cluster ^  
  --cache-cluster-id my-read-replica ^  
  --replication-group-id my-replication-group
```

Per ulteriori informazioni sull'aggiunta di una replica di lettura utilizzando la CLI, consulta [create-cache-cluster](#) nella Guida di riferimento alla riga di comando di Amazon ElastiCache

Aggiunta di una replica di lettura a un gruppo di replica utilizzando l'API

Per aggiungere una replica di lettura a un gruppo di replica Redis (modalità cluster disabilitata), utilizza l'operazione `CreateCacheCluster` di ElastiCache con il parametro `ReplicationGroupId` per specificare il gruppo di replica cui aggiungere il cluster (nodo).

L'esempio seguente consente di creare il cluster `myReadReplica` e di aggiungerlo al gruppo di replica `myReplicationGroup`. I tipi di nodi, i gruppi di parametri, i gruppi di sicurezza, la finestra di manutenzione e altre impostazioni per la replica di lettura sono identiche a quelle di altri nodi `myReplicationGroup`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=CreateCacheCluster  
  &CacheClusterId=myReadReplica
```

```
&ReplicationGroupId=myReplicationGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni sull'aggiunta di una replica di lettura utilizzando l'API, consulta [CreateCacheCluster](#) nella Guida di riferimento delle API di Amazon ElastiCache

Eliminazione di una replica di lettura, per i gruppi di replica Redis (modalità cluster disabilitata)

Le informazioni nel seguente argomento sono valide solo per gruppi di replica Redis (modalità cluster disabilitata) .

Se il traffico di lettura sul gruppo di replica Redis cambia, potrebbe essere necessario aggiungere o rimuovere repliche di lettura. La rimozione di un nodo da un gruppo di replica Redis (modalità cluster disabilitata) equivale all'eliminazione di un cluster, sebbene esistano delle limitazioni:

- Non puoi rimuovere il nodo primario da un gruppo di replica. Se desideri eliminare il nodo primario, esegui le seguenti operazioni:
 1. Promuovi una replica di lettura al nodo primario. Per ulteriori informazioni sulla promozione di una replica di lettura al nodo primario, consulta [Promozione di una replica di lettura sul primario, per gruppi di replica Redis \(modalità cluster disabilitata\)](#).
 2. Elimina il vecchio nodo primario. Per una limitazione su questo metodo, consulta il punto successivo.
- Se l'opzione Multi-AZ è abilitata su un gruppo di replica, non puoi rimuovere l'ultima replica di lettura dal gruppo di replica. In questo caso, esegui queste operazioni:
 1. Modifica il gruppo di replica disabilitando la funzione Multi-AZ. Per ulteriori informazioni, consulta [Modifica di un gruppo di replica](#).
 2. Elimina la replica di lettura.

Puoi rimuovere una replica di lettura da un gruppo di replica Redis (modalità cluster disabilitata) utilizzando la console ElastiCache, laAWS CLI per ElastiCache o l'API ElastiCache.

Per le indicazioni sull'eliminazione di un cluster da un gruppo di replica Redis, consulta la documentazione seguente:

- [Utilizzo di AWS Management Console](#)
- [Utilizzo di AWS CLI](#)
- [Utilizzo dell'API ElastiCache](#)
- [Ridimensionamento dei cluster in Redis \(modalità cluster attivata\)](#)
- [Riduzione del numero di repliche in uno shard](#)

Promozione di una replica di lettura sul primario, per gruppi di replica Redis (modalità cluster disabilitata)

Le informazioni nel seguente argomento sono valide solo per gruppi di replica Redis (modalità cluster disabilitata) .

Puoi promuovere una replica di lettura Redis (modalità cluster disabilitata) a quello primario utilizzando lo AWS Management Console, la AWS CLI o l'API ElastiCache. Non puoi promuovere una replica di lettura al primario mentre la funzione Multi-AZ con failover automatico è abilitata sul gruppo di replica Redis (modalità cluster disabilitata). Per promuovere una replica Redis (modalità cluster abilitata) al nodo primario su un gruppo di replica abilitato per Multi-AZ, esegui le seguenti operazioni:

1. Modifica il gruppo di replica per disabilitare Multi-AZ (questa operazione non richiede che tutti i cluster siano nella stessa zona di disponibilità). Per ulteriori informazioni, consulta [Modifica di un gruppo di replica](#).
2. Promuovere la replica di lettura al nodo primario.
3. Modifica il gruppo di replica per abilitare nuovamente Multi-AZ.

La funzione Multi-AZ non è disponibile nei gruppi di replica che eseguono Redis 2.6.13 o versioni precedenti.

Utilizzo di AWS Management Console

La procedura seguente utilizza la console per promuovere un nodo di replica al nodo primario.

Per promuovere una replica di lettura al nodo primario (console)

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Se la replica che si desidera promuovere è un membro di un gruppo di replica Redis (modalità cluster disabilitata) in cui Multi-AZ è abilitata, modificare il gruppo di replica per disabilitare Multi-AZ prima di continuare. Per ulteriori informazioni, consulta [Modifica di un gruppo di replica](#).
3. Scegliere Redis, quindi dall'elenco di cluster, scegliere il gruppo di replica che si desidera modificare. Questo gruppo di replica deve eseguire il motore "Redis", non il motore "Clustered Redis" e deve avere due o più nodi.
4. Dall'elenco di nodi, scegliere il nodo di replica che si desidera promuovere al nodo primario, quindi per Actions (Operazioni), scegliere Promote (Promuovi).

5. Nella finestra di dialogo Promote Read Replica (Promuovi replica di lettura), procedere come segue:
 - a. Per Apply Immediately (Applica immediatamente), scegliere Yes (Sì) per promuovere immediatamente la replica di lettura o No per promuoverla alla successiva finestra di manutenzione del cluster.
 - b. Scegliere Promote (Promuovi) per promuovere la replica di lettura o Cancel (Annulla) per annullare l'operazione.
6. Se il cluster è stato abilitato per Multi-AZ prima di iniziare il processo di promozione, attendere finché lo stato del gruppo di replica è available (disponibile), quindi modificare il cluster per abilitare nuovamente Multi-AZ. Per ulteriori informazioni, consulta [Modifica di un gruppo di replica](#).

Utilizzo di AWS CLI

Non puoi promuovere una replica di lettura al nodo primario se il gruppo di replica è abilitato per Multi-AZ. In alcuni casi, la replica che desideri promuovere potrebbe essere membro di un gruppo di repliche in cui è abilitata la funzione Multi-AZ. In questi casi, prima di continuare è necessario modificare il gruppo di replica per disabilitare Multi-AZ. Questa operazione non richiede che tutti i cluster siano nella stessa zona di disponibilità. Per ulteriori informazioni sulla modifica di un gruppo di replica, consulta [Modifica di un gruppo di replica](#).

Il comando AWS CLI seguente consente di modificare il gruppo di replica `sample-repl-group`, rendendo la replica di lettura `my-replica-1` il nodo primario nel gruppo di replica.

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id sample-repl-group \  
  --primary-cluster-id my-replica-1
```

Per Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id sample-repl-group ^  
  --primary-cluster-id my-replica-1
```

Per ulteriori informazioni sulla modifica di un gruppo di replica, consulta [modify-replication-group](#) nella Guida di riferimento alla riga di comando di Amazon ElastiCache

Utilizzo dell'API ElastiCache

Non puoi promuovere una replica di lettura al nodo primario se il gruppo di replica è abilitato per Multi-AZ. In alcuni casi, la replica che desideri promuovere potrebbe essere membro di un gruppo di repliche in cui è abilitata la funzione Multi-AZ. In questi casi, prima di continuare è necessario modificare il gruppo di replica per disabilitare Multi-AZ. Questa operazione non richiede che tutti i cluster siano nella stessa zona di disponibilità. Per ulteriori informazioni sulla modifica di un gruppo di replica, consulta [Modifica di un gruppo di replica](#).

L'operazione API ElastiCache seguente consente di modificare il gruppo di replica `myRep1Group`, rendendo la replica di lettura `myReplica-1` il nodo primario nel gruppo di replica.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyReplicationGroup  
&ReplicationGroupId=myRep1Group  
&PrimaryClusterId=myReplica-1  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Per ulteriori informazioni, consulta sulla modifica di un gruppo di replica, consulta [ModifyReplicationGroup](#) nell Guida di riferimento delle API di Amazon ElastiCache.

Gestione della manutenzione

Ogni cluster e gruppo di replica 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 o gruppo di replica, ElastiCache 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	Regione Asia Pacifico (Tokyo)	13:00 - 21:00 UTC
ap-northeast-2	Regione Asia Pacifico (Seoul)	12:00 - 20:00 UTC
ap-northeast-3	Regione Asia Pacifico (Osaka-Lo cale)	12:00 - 20:00 UTC
ap-southeast-3	Regione Asia Pacifico (Giacarta)	14:00 - 22:00 UTC
ap-south-1	Regione Asia Pacifico (Mumbai)	17:30-1:30 UTC
ap-southeast-1	Regione Asia Pacifico (Singapore)	14:00 - 22:00 UTC
cn-north-1	Regione Cina (Pechino)	14:00 - 22:00 UTC
cn-northwest-1	Regione Cina (Ningxia)	14:00 - 22:00 UTC
ap-east-1	Regione Asia Pacifico (Hong Kong)	13:00 - 21:00 UTC
ap-southeast-2	Asia Pacifico (Sydney)	12:00 - 20:00 UTC
eu-west-3	Regione UE (Parigi)	23:59 - 07:29 UTC
af-south-1	Regione Africa (Città del Capo)	13:00 - 21:00 UTC
eu-central-1	Regione Europa (Francoforte)	23:00 - 07:00 UTC
eu-west-1	Europa (Irlanda)	22:00 - 06:00 UTC
eu-west-2	Regione Europa (Londra)	23:00 - 07:00 UTC
me-south-1	Regione Medio Oriente (Bahrein)	13:00 - 21:00 UTC

Codice regione	Nome della regione	Finestra di manutenzione della regione
me-central-1	Regione Medio Oriente (Emirati Arabi Uniti)	13:00 - 21:00 UTC
eu-south-1	Regione Europa (Milano)	21:00 - 05:00 UTC
sa-east-1	Regione Sud America (San Paolo)	01:00 - 09:00 UTC
us-east-1	Stati Uniti orientali (Virginia settentrionale)	03:00 - 11:00 UTC
us-east-2	Stati Uniti orientali (Ohio)	04:00–12:00 UTC
us-gov-west-1	Regione AWS GovCloud (US)	06:00 - 14:00 UTC
us-west-1	Regione Stati Uniti occidentali (California settentrionale)	06:00 - 14:00 UTC
us-west-2	Stati Uniti occidentali (Oregon)	06:00 - 14:00 UTC

Modifica della finestra di manutenzione di un cluster o gruppo di replica

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 o gruppo di replica 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.

Note

Se si desidera applicare modifiche al tipo di nodo e/o aggiornamenti del motore immediatamente utilizzando AWS Management Console, seleziona il riquadro **Applica immediatamente**. Altrimenti queste modifiche verranno applicate durante la successiva finestra di manutenzione programmata. Per utilizzare l'API, consulta [modify-replication-group](#) o [modify-cache-cluster](#).

Ulteriori informazioni

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

- [Manutenzione ElastiCache](#)—Domande frequenti sulla manutenzione e la sostituzione dei nodi
- [Sostituzione dei nodi](#):-Gestione della sostituzione dei nodi
- [Modifica di un gruppo di replica](#): modifica della finestra di manutenzione di un gruppo di replica

Configurazione dei parametri di motore con i gruppi di parametri

Amazon ElastiCache utilizza i parametri per controllare le proprietà di runtime di nodi e 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 Redis](#).

Argomenti

- [Gestione dei parametri](#)
- [Livelli dei gruppi di parametri cache](#)
- [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 Memcached specifici](#)
- [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. Se il cluster esegue ad esempio Redis versione 3.2.10, puoi usare solo gruppi di parametri della famiglia Redis3.2, predefiniti o personalizzati.
- Se modifichi un gruppo di parametri del cluster, i valori di un parametro modificabile in base a condizioni devono essere gli stessi sia nel gruppo di parametri corrente sia in quello nuovo.
- Quando modifichi i parametri di un cluster, la modifica viene applicata al cluster immediatamente o dopo il riavvio dei nodi del cluster, con le eccezioni riportate a seguire. Ciò è valido se modifichi il gruppo dei parametri del cluster o un valore di parametro nel gruppo dei parametri del cluster. Per determinare quando viene applicata la modifica di un determinato parametro, consulta la colonna relativa all'applicazione delle modifiche nelle tabelle in [Parametri specifici di Redis](#).

Per ulteriori informazioni sul riavvio, consultare [Riavvio dei nodi](#).

Modificazioni dei parametri Redis (modalità cluster abilitata)

Se apporti modifiche ai seguenti parametri in un cluster Redis (modalità cluster abilitata), segui i passaggi seguenti.

- `activeresharding`
 - `database`
1. Creare un backup manuale del cluster. Per informazioni, consultare [Esecuzione di backup manuali](#).
 2. Eliminare il cluster Redis (modalità cluster abilitata). Consulta [Deleting clusters](#).

3. Ripristina il cluster utilizzando il gruppo di parametri modificato e il backup per inizializzare il nuovo cluster. Per informazioni, consultare [Ripristino da un backup in una nuova cache](#).

Le modifiche ad altri parametri non richiedono questo.

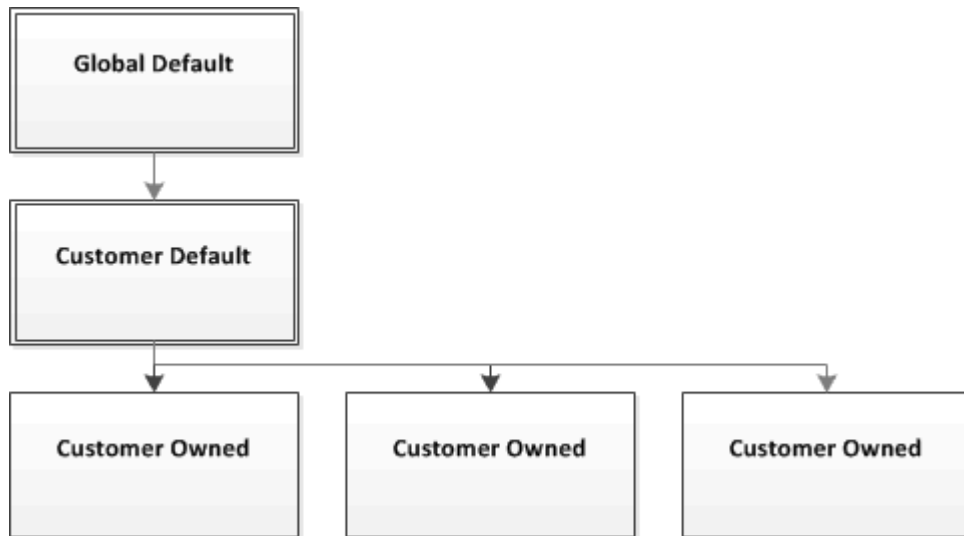
- È possibile associare gruppi di parametri ai Global Datastore Redis. I Datastore globali sono una raccolta di uno o più cluster che comprendono più regioni AWS. In questo caso, il gruppo di parametri viene condiviso da tutti i cluster che costituiscono il Global Datastore. Eventuali modifiche al gruppo di parametri del cluster primario vengono replicate in tutti i cluster rimanenti nel Global Datastore. Per ulteriori informazioni, consulta [Replica tra regioni AWS mediante Global Datastore](#).

È possibile verificare se un gruppo di parametri fa parte di un Global Datastore esaminando le seguenti posizioni:

- Nella console ElastiCache nella pagina Parameter Groups (Gruppi di parametri), con/senza l'attributo Global (Globale)
- Con/senza la proprietà `IsGlobal` dell'operazione API [CacheParameterGroup](#)

Livelli dei gruppi di parametri cache

Amazon ElastiCache ha tre livelli di gruppi di parametri cache come illustrato di seguito.



Livelli dei gruppi di parametri Amazon ElastiCache

Di default globale

Gruppo di parametri root di livello superiore per tutti i clienti di Amazon ElastiCache nella regione.

Gruppo di parametri cache di default globale:

- Riservato per ElastiCache e non disponibile per il cliente.

Di default del cliente

Copia del gruppo di parametri cache di default globale creata per l'uso da parte del cliente.

Il gruppo di parametri cache di default globale:

- Viene creato da ed è di ElastiCache.
- È disponibile per l'uso da parte del cliente come gruppo di parametri cache per i cluster che eseguono una versione del motore supportata dal gruppo di parametri cache.
- Non può essere modificato dal cliente.

Di proprietà del cliente

Copia del gruppo di parametri cache di default globale, un gruppo di parametri cache di proprietà del cliente viene creato ogni volta che il cliente crea un gruppo di parametri cache.

Il gruppo di parametri cache di proprietà del cliente:

- 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 cache 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. Puoi creare un gruppo di parametri utilizzando la console ElastiCache, il AWS CLI o l'API ElastiCache.

Creazione di un gruppo di parametri (Console)

La procedura seguente mostra come creare un gruppo di parametri tramite la console ElastiCache.

Per creare un gruppo di parametri tramite la console ElastiCache

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
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 nuovo gruppo di parametri, scegliere Create Parameter Group (Crea gruppo di parametri).

Viene visualizzata la schermata Create Parameter Group (Crea gruppo di parametri).

4. Nell'elenco Family (Famiglia) scegliere la famiglia del gruppo di parametri che verrà utilizzata come modello per il gruppo di parametri.

La famiglia del gruppo di parametri, ad esempio redis3.2, definisce i parametri effettivi nel gruppo di parametri e i relativi valori iniziali. e deve coincidere con la versione e il motore del cluster.

5. 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.
6. Nella casella Description (Descrizione) digitare una descrizione per il gruppo di parametri.
 7. Per creare il gruppo di parametri, scegliere Create (Crea).

Per terminare il processo senza creare il gruppo di parametri, scegliere Cancel (Annulla).

8. 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 tramite AWS CLI, usa il comando `create-cache-parameter-group` con questi parametri.

- `--cache-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.
- `--cache-parameter-group-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 nuovo gruppo di parametri denominato `myRed28`, che utilizza la famiglia `redis2.8` come modello.

Per Linux, macOS o Unix:

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name myRed28 \  
  --cache-parameter-group-family redis2.8 \  
  --description "My first parameter group"
```

Per Windows:

```
aws elasticache create-cache-parameter-group ^  
  --cache-parameter-group-name myRed28 ^  
  --cache-parameter-group-family redis2.8 ^  
  --description "My first parameter group"
```

L'output di questo comando dovrebbe essere simile a quanto segue:

```
{  
  "CacheParameterGroup": {  
    "CacheParameterGroupName": "myRed28",  
    "CacheParameterGroupFamily": "redis2.8",  
    "Description": "My first parameter group"  
  }  
}
```

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, consulta [create-cache-parameter-group](#).

Creazione di un gruppo di parametri (API ElastiCache)

Per creare un gruppo di parametri tramite l'API ElastiCache, usa l'operazione `CreateCacheParameterGroup` 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.
- `CacheParameterGroupFamily`— Il motore e la famiglia di versioni per il gruppo di parametri. Ad esempio, `redis2.8`.
- `Description` - Una descrizione per la copia del gruppo di parametri del cluster fornita dall'utente.

Example

L'esempio seguente crea un nuovo gruppo di parametri denominato `myRed28`, che utilizza la famiglia `redis2.8` come modello.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateCacheParameterGroup  
&CacheParameterGroupFamily=redis2.8  
&CacheParameterGroupName=myRed28  
&Description=My%20first%20parameter%20group  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

La risposta restituita da tale operazione dovrebbe essere simile a quanto segue:

```
<CreateCacheParameterGroupResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <CreateCacheParameterGroupResult>  
    <CacheParameterGroup>
```



```
<CacheParameterGroupName>myRed28</CacheParameterGroupName>
<CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
<Description>My first parameter group</Description>
</CacheParameterGroup>
</CreateCacheParameterGroupResult>
<ResponseMetadata>
  <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>
</ResponseMetadata>
</CreateCacheParameterGroupResponse>
```

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, consulta [CreateCacheParameterGroup](#).

Elenco di gruppi di parametri per nome

Puoi elencare i gruppi di parametri utilizzando la console ElastiCache, la AWS CLI o l'API ElastiCache.

Elenco di gruppi di parametri per nome (console)

La procedura seguente mostra come visualizzare un elenco di gruppi di parametri tramite la console ElastiCache.

Per elencare i gruppi di parametri tramite la console ElastiCache

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Per un elenco di tutti i gruppi di parametri disponibili, nel pannello di navigazione a sinistra scegliere Parameter Groups (Gruppi di parametri).

Elenco di gruppi di parametri per nome (AWS CLI)

Per generare un elenco di gruppi di parametri tramite AWS CLI, utilizza il comando `describe-cache-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-records` 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 `myRed28`.

Per Linux, macOS o Unix:

```
aws elasticache describe-cache-parameter-groups \  
  --cache-parameter-group-name myRed28
```

Per Windows:

```
aws elasticache describe-cache-parameter-groups ^  
  --cache-parameter-group-name myRed28
```

L'output di questo comando sarà simile al seguente e conterrà il nome, la famiglia e la descrizione del gruppo di parametri.

```
{
  "CacheParameterGroups": [
    {
      "CacheParameterGroupName": "myRed28",
      "CacheParameterGroupFamily": "redis2.8",
      "Description": "My first parameter group"
    }
  ]
}
```

Example

Il seguente codice di esempio elenca il gruppo di parametri myRed56 per i gruppi di parametri in esecuzione sul motore Redis versione 5.0.6 e versioni successive. Se il gruppo di parametri fa parte di un [Replica tra regioni AWS mediante Global Datastore](#), il valore della proprietà `IsGlobal` restituito nell'output sarà `Yes`.

Per Linux, macOS o Unix:

```
aws elasticache describe-cache-parameter-groups \
  --cache-parameter-group-name myRed56
```

Per Windows:

```
aws elasticache describe-cache-parameter-groups ^
  --cache-parameter-group-name myRed56
```

L'output di questo comando sarà simile al seguente e conterrà il nome, la famiglia, `isGlobal` e la descrizione del gruppo di parametri.

```
{
  "CacheParameterGroups": [
    {
      "CacheParameterGroupName": "myRed56",
      "CacheParameterGroupFamily": "redis5.0",
      "Description": "My first parameter group",
      "IsGlobal": "yes"
    }
  ]
}
```

```
    }  
  ]  
}
```

Example

Il codice di esempio seguente elenca fino a 10 gruppi di parametri.

```
aws elasticache describe-cache-parameter-groups --max-records 10
```

L'output JSON di questo comando sarà simile a questo, elencando per ogni gruppo di parametri il nome, la famiglia, la descrizione e, nel caso di redis5.6, se il gruppo di parametri fa parte di un Global Datastore (isGlobal).

```
{  
  "CacheParameterGroups": [  
    {  
      "CacheParameterGroupName": "custom-redis32",  
      "CacheParameterGroupFamily": "redis3.2",  
      "Description": "custom parameter group with reserved-memory > 0"  
    },  
    {  
      "CacheParameterGroupName": "default.memcached1.4",  
      "CacheParameterGroupFamily": "memcached1.4",  
      "Description": "Default parameter group for memcached1.4"  
    },  
    {  
      "CacheParameterGroupName": "default.redis2.6",  
      "CacheParameterGroupFamily": "redis2.6",  
      "Description": "Default parameter group for redis2.6"  
    },  
    {  
      "CacheParameterGroupName": "default.redis2.8",  
      "CacheParameterGroupFamily": "redis2.8",  
      "Description": "Default parameter group for redis2.8"  
    },  
    {  
      "CacheParameterGroupName": "default.redis3.2",  
      "CacheParameterGroupFamily": "redis3.2",  
      "Description": "Default parameter group for redis3.2"  
    },  
    {
```

```
    "CacheParameterGroupName": "default.redis3.2.cluster.on",
    "CacheParameterGroupFamily": "redis3.2",
    "Description": "Customized default parameter group for redis3.2 with
cluster mode on"
  },
  {
    "CacheParameterGroupName": "default.redis5.6.cluster.on",
    "CacheParameterGroupFamily": "redis5.0",
    "Description": "Customized default parameter group for redis5.6 with
cluster mode on",
    "isGlobal": "yes"
  },
]
}
```

Per ulteriori informazioni, consulta [describe-cache-parameter-groups](#).

Elenco dei gruppi di parametri per nome (API ElastiCache)

Per generare un elenco di gruppi di parametri tramite l'API ElastiCache, utilizza l'operazione `DescribeCacheParameterGroups`. 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 `MaxRecords` 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 10 gruppi di parametri.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&MaxRecords=10
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

La risposta di questa operazione sarà simile a questa, elencando per ogni gruppo di parametri il nome, la famiglia, la descrizione e, nel caso di redis5.6 se il gruppo di parametri appartiene a un Global Datastore (`isGlobal`).

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myRedis28</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
        <Description>My custom Redis 2.8 parameter group</Description>
      </CacheParameterGroup>
      <CacheParameterGroup>
        <CacheParameterGroupName>myMem14</CacheParameterGroupName>
        <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
        <Description>My custom Memcached 1.4 parameter group</Description>
      </CacheParameterGroup>
      <CacheParameterGroup>
        <CacheParameterGroupName>myRedis56</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>
        <Description>My custom redis 5.6 parameter group</Description>
        <isGlobal>yes</isGlobal>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

Example

Il codice di esempio seguente elenca il gruppo di parametri myRed28.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&CacheParameterGroupName=myRed28
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

La risposta restituita da tale operazione sarà simile a quanto segue e conterrà il nome, la famiglia e la descrizione.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myRed28</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
        <Description>My custom Redis 2.8 parameter group</Description>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

Example

Il codice di esempio seguente elenca il gruppo di parametri myRed56.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&CacheParameterGroupName=myRed56
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

La risposta di questa operazione sarà simile a questa, elencando il nome, la famiglia, la descrizione e se il gruppo di parametri fa parte di un Datastore globale (isGlobal).

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myRed56</CacheParameterGroupName>
        <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>
        <Description>My custom Redis 5.6 parameter group</Description>
        <isGlobal>yes</isGlobal>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
</DescribeCacheParameterGroupsResponse>
```

```
</CacheParameterGroups>
</DescribeCacheParameterGroupsResult>
<ResponseMetadata>
  <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
</ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

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

Generazione di un elenco di valori di un gruppo di parametri

Puoi creare un elenco di parametri e dei relativi valori di un gruppo di parametri tramite la console ElastiCache, la AWS CLI o l'API ElastiCache.

Generazione di un elenco di valori di un gruppo di parametri (console)

La procedura seguente mostra come creare un elenco dei parametri e dei relativi valori per un gruppo di parametri tramite la console ElastiCache.

Per creare un elenco dei parametri e dei relativi valori in un gruppo di parametri tramite la console ElastiCache.

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Per un elenco di tutti i gruppi di parametri disponibili, nel pannello di navigazione a sinistra scegliere Parameter Groups (Gruppi di parametri).
3. Scegliere il gruppo di parametri per cui si desidera elencare i parametri e i valori selezionando la casella a sinistra 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.

Generazione di un elenco di valori di un gruppo di parametri (AWS CLI)

Per creare un elenco dei parametri e dei relativi valori in un gruppo di parametri tramite AWS CLI, utilizza il comando `describe-cache-parameters`.

Example

Il seguente codice di esempio consente di creare un elenco di tutti i parametri e dei relativi valori per il gruppo di parametri `myRedis28`.

Per Linux, macOS o Unix:

```
aws elasticache describe-cache-parameters \  
  --cache-parameter-group-name myRedis28
```

Per Windows:

```
aws elasticache describe-cache-parameters ^  
  --cache-parameter-group-name myRed28
```

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

Generazione di un elenco di valori di un gruppo di parametri (API ElastiCache)

Per creare un elenco dei parametri e dei relativi valori in un gruppo di parametri tramite l'API ElastiCache, utilizza l'operazione DescribeCacheParameters.

Example

Il seguente codice di esempio consente di creare un elenco di tutti i parametri per il gruppo di parametri myRed28.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheParameters  
  &CacheParameterGroupName=myRed28  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &Version=2015-02-02  
  &X-Amz-Credential=<credential>
```

La risposta restituita da tale operazione dovrebbe essere simile a quanto segue. In questo caso, la risposta è stata troncata.

```
<DescribeCacheParametersResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <DescribeCacheParametersResult>  
    <CacheClusterClassSpecificParameters>  
      <CacheNodeTypeSpecificParameter>  
        <DataType>integer</DataType>  
        <Source>system</Source>  
        <IsModifiable>>false</IsModifiable>  
        <Description>The maximum configurable amount of memory to use to store items,  
in megabytes.</Description>  
        <CacheNodeTypeSpecificValues>  
          <CacheNodeTypeSpecificValue>  
            <Value>1000</Value>  
            <CacheClusterClass>cache.c1.medium</CacheClusterClass>
```

```
</CacheNodeTypeSpecificValue>
<CacheNodeTypeSpecificValue>
  <Value>6000</Value>
  <CacheClusterClass>cache.c1.xlarge</CacheClusterClass>
</CacheNodeTypeSpecificValue>
<CacheNodeTypeSpecificValue>
  <Value>7100</Value>
  <CacheClusterClass>cache.m1.large</CacheClusterClass>
</CacheNodeTypeSpecificValue>
<CacheNodeTypeSpecificValue>
  <Value>1300</Value>
  <CacheClusterClass>cache.m1.small</CacheClusterClass>
</CacheNodeTypeSpecificValue>
```

...output omitted...

```
</CacheClusterClassSpecificParameters>
</DescribeCacheParametersResult>
<ResponseMetadata>
  <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
</ResponseMetadata>
</DescribeCacheParametersResponse>
```

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

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 `cluster-enabled` tramite la console ElastiCache. Puoi usare la stessa procedura per modificare il valore di qualsiasi parametro.

Per modificare il valore di un parametro tramite la console ElastiCache

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Per un elenco di tutti i gruppi di parametri disponibili, nel pannello di navigazione a sinistra scegliere Parameter Groups (Gruppi di parametri).
3. Scegliere il gruppo di parametri che si desidera modificare selezionando la casella a sinistra del nome del gruppo di parametri.

I parametri del gruppo di parametri verranno elencati nella parte inferiore dello schermo.

Potrebbe essere necessario scorrere l'elenco per visualizzare tutti i parametri.

4. Per modificare uno o più parametri, scegliere Edit Parameters (Modifica parametri).
5. Seleziona Salva modifiche.
6. Per trovare il nome del parametro modificato, consultare [Parametri specifici di Redis](#). Se si dispone di un cluster Redis (modalità cluster disabilitata) e si apportano modifiche ai seguenti parametri, è necessario riavviare i nodi nel cluster:

- activerehashing
- database

Per ulteriori informazioni sul riavvio, consultare [Riavvio dei nodi](#).

Modificazioni dei parametri Redis (modalità cluster abilitata)

Se apporti modifiche ai seguenti parametri in un cluster Redis (modalità cluster abilitata), segui i passaggi seguenti.

- activerehashing
- database

1. Creare un backup manuale del cluster. Per informazioni, consultare [Esecuzione di backup manuali](#).
2. Eliminare il cluster Redis (modalità cluster abilitata). Consulta [Deleting clusters](#).
3. Ripristinare il cluster utilizzando il gruppo di parametri modificato e il backup per inizializzare il nuovo cluster. Per informazioni, consultare [Ripristino da un backup in una nuova cache](#).

Le modifiche ad altri parametri non richiedono questo.

Modifica di un gruppo di parametri (AWS CLI)

Per modificare il valore di un parametro tramite AWS CLI, utilizza il comando `modify-cache-parameter-group`.

Example

Per individuare il nome e i valori consentiti del parametro da modificare, consulta [Parametri specifici di Redis](#)

Il seguente codice di esempio imposta il valore di due parametri, `reserved-memory-percent` e `cluster-enabled`, nel gruppo di parametri `myredis32-on-30`. Impostiamo `reserved-memory-percent` su 30 (30 per cento) e `cluster-enabled` su `yes` in modo tale che il gruppo di parametri possa essere utilizzato con i cluster (gruppi di replica).

Per Linux, macOS o Unix:

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name myredis32-on-30 \  
  --parameter-name-values \  
    ParameterName=reserved-memory-percent,ParameterValue=30 \  
    ParameterName=cluster-enabled,ParameterValue=yes
```

Per Windows:

```
aws elasticache modify-cache-parameter-group ^  
  --cache-parameter-group-name myredis32-on-30 ^  
  --parameter-name-values ^  
    ParameterName=reserved-memory-percent,ParameterValue=30 ^  
    ParameterName=cluster-enabled,ParameterValue=yes
```

L'aspetto dell'output di questo comando sarà simile al seguente.

```
{
```

```
"CacheParameterGroupName": "my-redis32-on-30"  
}
```

Per ulteriori informazioni, consulta [modify-cache-parameter-group](#).

Per trovare il nome del parametro modificato, consultare [Parametri specifici di Redis](#).

Se si dispone di un cluster Redis (modalità cluster disabilitata) e si apportano modifiche ai seguenti parametri, è necessario riavviare i nodi nel cluster:

- activerehashing
- database

Per ulteriori informazioni sul riavvio, consultare [Riavvio dei nodi](#).

Modificazioni dei parametri Redis (modalità cluster abilitata)

Se apporti modifiche ai seguenti parametri in un cluster Redis (modalità cluster abilitata), segui i passaggi seguenti.

- activerehashing
 - database
1. Creare un backup manuale del cluster. Per informazioni, consultare [Esecuzione di backup manuali](#).
 2. Eliminare il cluster Redis (modalità cluster abilitata). Consulta [Deleting clusters](#).
 3. Ripristinare il cluster utilizzando il gruppo di parametri modificato e il backup per inizializzare il nuovo cluster. Per informazioni, consultare [Ripristino da un backup in una nuova cache](#).

Le modifiche ad altri parametri non richiedono questo.

Modifica di un gruppo di parametri (API ElastiCache)

Per modificare i valori dei parametri di un gruppo di parametri tramite l'API ElastiCache, utilizza l'operazione `ModifyCacheParameterGroup`.

Example

Per individuare il nome e i valori consentiti del parametro da modificare, consulta [Parametri specifici di Redis](#)

Il seguente codice di esempio imposta il valore di due parametri, `reserved-memory-percent` e `cluster-enabled`, nel gruppo di parametri `myredis32-on-30`. Impostiamo `reserved-memory-percent` su `30` (30 per cento) e `cluster-enabled` su `yes` in modo tale che il gruppo di parametri possa essere utilizzato con i cluster (gruppi di replica).

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheParameterGroup  
&CacheParameterGroupName=myredis32-on-30  
&ParameterNameValues.member.1.ParameterName=reserved-memory-percent  
&ParameterNameValues.member.1.ParameterValue=30  
&ParameterNameValues.member.2.ParameterName=cluster-enabled  
&ParameterNameValues.member.2.ParameterValue=yes  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

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

Se si dispone di un cluster Redis (modalità cluster disabilitata) e si apportano modifiche ai seguenti parametri, è necessario riavviare i nodi nel cluster:

- `activeresharding`
- `database`

Per ulteriori informazioni sul riavvio, consultare [Riavvio dei nodi](#).

Modificazioni dei parametri Redis (modalità cluster abilitata)

Se apporti modifiche ai seguenti parametri in un cluster Redis (modalità cluster abilitata), segui i passaggi seguenti.

- `activeresharding`
- `database`

1. Creare un backup manuale del cluster. Per informazioni, consultare [Esecuzione di backup manuali](#).
2. Eliminare il cluster Redis (modalità cluster abilitata). Per informazioni, consultare [Eliminazione di un cluster](#).
3. Ripristinare il cluster utilizzando il gruppo di parametri modificato e il backup per inizializzare il nuovo cluster. Per informazioni, consultare [Ripristino da un backup in una nuova cache](#).

Le modifiche ad altri parametri non richiedono questo.

Eliminazione di un gruppo di parametri

Puoi creare un gruppo di parametri utilizzando la console ElastiCache, la AWS CLI o l'API ElastiCache.

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 tramite la console ElastiCache.

Per eliminare un gruppo di parametri tramite la console ElastiCache

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Per un elenco di tutti i gruppi di parametri disponibili, nel pannello di navigazione a sinistra scegliere Parameter Groups (Gruppi di parametri).
3. Scegliere i gruppi di parametri che si desidera eliminare selezionando la casella a sinistra del nome del gruppo di parametri.

Il pulsante Delete (Elimina) diventa attivo.

4. Scegliere Delete (Elimina).

Verrà visualizzata la schermata di conferma Delete Parameter Groups (Elimina gruppi di parametri).

5. Per eliminare i gruppi di parametri, nella schermata di conferma Delete Parameter Groups (Elimina gruppi di parametri) scegliere Delete (Elimina).

Per mantenere i gruppi di parametri, scegliere Cancel (Annulla).

Eliminazione di un gruppo di parametri (AWS CLI)

Per eliminare un gruppo di parametri tramite AWS CLI, utilizza il comando `delete-cache-parameter-group`. Per il gruppo di parametri da eliminare, il gruppo di parametri specificato da `--cache-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 `myMem14`.

Example

Per Linux, macOS o Unix:

```
aws elasticache delete-cache-parameter-group \  
  --cache-parameter-group-name myRed28
```

Per Windows:

```
aws elasticache delete-cache-parameter-group ^  
  --cache-parameter-group-name myRed28
```

Per ulteriori informazioni, consulta [delete-cache-parameter-group](#).

Eliminazione di un gruppo di parametri (API ElastiCache)

Per eliminare un gruppo di parametri tramite l'API ElastiCache, utilizza l'operazione `DeleteCacheParameterGroup`. Per il gruppo di parametri da eliminare, il gruppo di parametri specificato da `CacheParameterGroupName` 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 `myRed28`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DeleteCacheParameterGroup  
  &CacheParameterGroupName=myRed28  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &Version=2015-02-02  
  &X-Amz-Credential=<credential>
```

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

Parametri Memcached specifici

Se non specifichi un gruppo di parametri per il cluster Memcached, verrà utilizzato un gruppo di parametri di default appropriato per la versione del motore. Non è possibile modificare nessun valore all'interno del gruppo di parametri di default. Tuttavia, è possibile creare un gruppo di parametri personalizzato e assegnarlo al cluster in qualsiasi momento. Per ulteriori informazioni, consulta [Creazione di un gruppo di parametri](#).

Argomenti

- [Modifiche di Memcached 1.6.17](#)
- [Parametri di Memcached 1.6.6 aggiunti](#)
- [Modifiche dei parametri di Memcached 1.5.10](#)
- [Parametri di Memcached 1.4.34 aggiunti](#)
- [Parametri di Memcached 1.4.33 aggiunti](#)
- [Parametri di Memcached 1.4.24 aggiunti](#)
- [Parametri di Memcached 1.4.14 aggiunti](#)
- [Parametri di Memcached 1.4.5 supportati](#)
- [Sovraccarico delle connessioni Memcached](#)
- [Parametri specifici del tipo di nodo Memcached](#)

Modifiche di Memcached 1.6.17

A partire da Memcached 1.6.17 non sono più supportati questi comandi amministrativi: `lru_crawler`, `lru` e `slabs`. Con queste modifiche non sarai in grado di abilitare/disabilitare `lru_crawler` in fase di runtime usando i comandi. Abilita/disabilita `lru_crawler` modificando il tuo gruppo di parametri personalizzato.

Parametri di Memcached 1.6.6 aggiunti


Per Memcached 1.6.6, non sono supportati parametri aggiuntivi.

Famiglia del gruppo di parametri: `memcached1.6`

Modifiche dei parametri di Memcached 1.5.10

Per Memcached 1.5.10, sono supportati i seguenti parametri aggiuntivi.

Famiglia del gruppo di parametri: `memcached1.5`

Nome	Informazioni	Descrizione
no_modern	<p>Impostazione predefinita: 1</p> <p>Tipo: booleano</p> <p>Modificabile: sì</p> <p>Valori consentiti: 0,1</p> <p>Modificazioni hanno effetto: All'avvio</p>	<p>Un alias per disabilitare i comandi <code>slab_reassign</code> , <code>slab_automove</code> , <code>lru_crawler</code> , <code>lru_maintainer</code> , <code>maxconns_fast</code> . No modern imposta anche l'algoritmo hash su <code>jenkins</code> e consente l'inlining di ASCII VALUE. Applicabile a memcached 1.5 e versioni successive. Per tornare al moderno, è necessario disabilitare questo parametro e riavviare , che abilita automaticamente <code>slab_reassign</code> , <code>slab_automove</code> , <code>lru_crawler</code> , <code>lru_maintainer</code> e <code>maxconns_fast</code> .</p> <div data-bbox="1008 1171 1511 1885"><p> Note</p><p>Il valore di configurazione di default per questo parametro è stato modificato da 0 a 1 a partire dal 20 agosto 2021. Il valore di default aggiornato verrà automaticamente rilevato dai nuovi utenti di Elasticache per ciascuna regione dopo il 20 agosto 2021. Gli utenti di ElastiCache esistenti nelle regioni prima del</p></div>

Nome	Informazioni	Descrizione
		20 agosto 2021 devono modificare manualmente i gruppi di parametri personalizzati per poter rilevare questa nuova modifica.
<code>inline_ascii_resp</code>	Impostazione Predefinita: 0 Tipo: booleano Modificabile: sì Valori consentiti: 0,1 Modificazioni hanno effetto: Al lancio	Archivia i numeri dalla risposta VALUE in una voce usando fino a 24 byte. Lieve rallentamento dei set ASCII get, faster.

Per Memcached 1.5.10, i seguenti parametri sono stati rimossi.

Nome	Informazioni	Descrizione
<code>expirezero_does_no_t_evict</code>	Impostazione Predefinita: 0 Tipo: booleano Modificabile: sì Valori consentiti: 0,1 Modificazioni hanno effetto: Al lancio	Non più supportati in questa versione.
<code>modern</code>	Impostazione predefinita: 1	

Nome	Informazioni	Descrizione
	Tipo: booleano Modificabile: Sì (richiede il riavvio, se impostato su <code>no_modern</code>) Valori consentiti: 0,1 Modificazioni hanno effetto: Al lancio	Non più supportati in questa versione. A partire da questa versione <code>no-modern</code> è abilitato per impostazione predefinita a ogni avvio o riavvio.

Parametri di Memcached 1.4.34 aggiunti

Per Memcached 1.4.34, non sono supportati parametri aggiuntivi.

Famiglia del gruppo di parametri: `memcached1.4`

Parametri di Memcached 1.4.33 aggiunti

Per Memcached 1.4.33, sono supportati i seguenti parametri aggiuntivi.

Famiglia del gruppo di parametri: `memcached1.4`

Nome	Informazioni	Descrizione
<code>modern</code>	Impostazione predefinita: abilitato Tipo: booleano Modificabile: sì Modificazioni hanno effetto: Al lancio	Un alias per più funzionalità. L'abilitazione di <code>modern</code> equivale all'attivazione dei seguenti comandi e all'utilizzo dell'algoritmo hash murmur3: <code>slab_reassign</code> , <code>slab_automove</code> , <code>lru_crawler</code> , <code>lru_maintainer</code> , <code>maxconns_fast</code> e <code>hash_algorithm=murmur3</code> .

Nome	Informazioni	Descrizione
<code>watch</code>	<p>Impostazione predefinita: abilitato</p> <p>Tipo: booleano</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p> <p>I log possono essere eliminati se l'utente raggiunge i limiti relativi a <code>watcher_logbuf_size</code> e <code>worker_logbuf_size</code>.</p>	<p>Recupero, espulsione o mutazione di log. Ad esempio, quando un utente attiva <code>watch</code>, può visualizzare i log quando si verifica <code>get</code>, <code>set</code>, <code>delete</code> o <code>update</code>.</p>
<code>idle_timeout</code>	<p>Impostazione predefinita: 0 (disabilitato)</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: Al lancio</p>	<p>Il numero minimo di secondi durante i quali un client può essere inattivo prima che ne venga richiesta la chiusura. Intervallo di valori: da 0 a 86400.</p>

Nome	Informazioni	Descrizione
<code>track_sizes</code>	<p>Impostazione predefinita: disabilitato</p> <p>Tipo: booleano</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: Al lancio</p>	<p>Mostra le dimensioni utilizzate da ciascun gruppo di slab.</p> <p>L'abilitazione di <code>track_sizes</code> consente di eseguire <code>stats sizes</code> senza dover eseguire <code>stats sizes_enable</code>.</p>
<code>watcher_logbuf_size</code>	<p>Impostazione predefinita: 256 (KB)</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: Al lancio</p>	<p>Il comando <code>watch</code> abilita la registrazione del flusso per Memcached. <code>watch</code> può tuttavia eliminare i log se il tasso di espulsioni, mutazioni o recuperi è sufficientemente elevato per riempire il buffer di registrazione. In tali situazioni, gli utenti possono aumentare le dimensioni del buffer in modo da ridurre la possibile perdita di log.</p>
<code>worker_logbuf_size</code>	<p>Impostazione predefinita: 64 (KB)</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: Al lancio</p>	<p>Il comando <code>watch</code> abilita la registrazione del flusso per Memcached. <code>watch</code> può tuttavia eliminare i log se il tasso di espulsioni, mutazioni o recuperi è sufficientemente elevato per riempire il buffer di registrazione. In tali situazioni, gli utenti possono aumentare le dimensioni del buffer in modo da ridurre la possibile perdita di log.</p>

Nome	Informazioni	Descrizione
<code>slab_chunk_max</code>	<p>Impostazione predefinita: 524288 (byte)</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: Al lancio</p>	<p>Specifica le dimensioni massime di uno slab. Impostando una dimensione inferiore di slab viene utilizzata una memoria più efficiente. Gli elementi più grandi di <code>slab_chunk_max</code> sono suddivisi in più slab.</p>
<code>lru_crawler metadump [all 1 2 3]</code>	<p>Impostazione predefinita: disabilitato</p> <p>Tipo: booleano</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Se <code>lru_crawler</code> è abilitato questo comando effettua il dump di tutte le chiavi.</p> <p><code>all 1 2 3</code> : tutti gli slab o specificare un determinato numero di slab</p>



Parametri di Memcached 1.4.24 aggiunti


Per Memcached 1.4.24, sono supportati i seguenti parametri aggiuntivi.

Famiglia del gruppo di parametri: memcached1.4

Nome	Informazioni	Descrizione
<code>disable_flush_all</code>	<p>Impostazione predefinita: 0 (disabilitato)</p> <p>Tipo: booleano</p> <p>Modificabile: sì</p>	<p>Aggiungere il parametro (-F) per disabilitare <code>flush_all</code>. Utile se non desideri poter eseguire uno svuotamento completo sulle istanze di produzione.</p>

Nome	Informazioni	Descrizione
	Modificazioni hanno effetto: Al lancio	Valori: 0, 1 (l'utente può utilizzare il parametro <code>flush_all</code> quando il valore è 0).
<code>hash_algorithm</code>	Impostazione predefinita: jenkins Tipo: stringa Modificabile: sì Modificazioni hanno effetto: Al lancio	L'algoritmo hash da utilizzare. Valori consentiti: murmur3 e jenkins.

Nome	Informazioni	Descrizione
<code>lru_crawler</code>	<p>Impostazione predefinita: 0 (disabilitato)</p> <p>Tipo: booleano</p> <p>Modificabile: sì</p> <p>Le modifiche hanno effetto: Dopo il riavvio</p> <div data-bbox="651 684 971 1430"><p> Note</p><p>Puoi abilitare temporaneamente <code>lru_crawler</code> in fase di runtime dalla riga di comando. Per ulteriori informazioni, consulta la colonna Descrizione.</p></div>	<p>Pulisce le classi di slab degli elementi scaduti. Si tratta di un processo a basso impatto che viene eseguito in background. Attualmente richiede l'avvio di un crawling tramite l'utilizzo di un comando manuale.</p> <p>Per un'abilitazione temporanea, esegui il comando <code>lru_crawler enable</code> alla riga di comando.</p> <p><code>lru_crawler 1,3,5</code> esegue il crawling delle classi slab 1, 3 e 5 per cercare elementi scaduti da aggiungere all'elenco libero.</p> <p>Valori: 0,1</p> <div data-bbox="1008 1100 1507 1799"><p> Note</p><p>L'abilitazione di <code>lru_crawler</code> alla riga di comando abilita il crawler finché non viene disabilitato alla riga di comando o al successivo riavvio. Per abilitarlo in modo permanente, devi modificarne il valore del parametro. Per ulteriori informazioni, consulta Modifica di un gruppo di parametri.</p></div>

Nome	Informazioni	Descrizione
<code>lru_maintainer</code>	Impostazione predefinita: 0 (disabilitato) Tipo: booleano Modificabile: sì Modificazioni hanno effetto: Al lancio	Un thread in background che distribuisce gli elementi utilizzati meno di recente man mano che si raggiungono le capacità. Valori: 0, 1.
<code>expirezero_does_not_evict</code>	Impostazione predefinita: 0 (disabilitato) Tipo: booleano Modificabile: sì Modificazioni hanno effetto: Al lancio	Quando utilizzato con <code>lru_maintainer</code> , rende gli elementi non soggetti a espulsione con un tempo di scadenza pari a 0. <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Warning Ciò consente di togliere spazio di memoria disponibile per gli elementi soggetti a espulsione.</p> </div> <p>Può essere impostato in modo da ignorare <code>lru_maintainer</code>.</p>

Parametri di Memcached 1.4.14 aggiunti

Per Memcached 1.4.14, sono supportati i seguenti parametri aggiuntivi.

Famiglia del gruppo di parametri: `memcached1.4`

Parametri aggiunti in Memcached 1.4.14

Nome	Descrizione
<code>config_max</code>	Il numero massimo di voci di configurazione ElastiCache.
<code>config_size_max</code>	Dimensione massima delle voci di configurazione, in byte.
<code>hashpower_init</code>	La dimensione iniziale della tabella hash ElastiCache, espressa come potenza di due. L'impostazione 16 (2^{16}) o 65536 chiavi.

Nome	Descrizione
maxconns_fast	<p>Modifica il modo in cui vengono gestite le nuove richieste di connessione quando viene raggiunto il limite massimo di connessioni. Se questo parametro è impostato su 0 (zero), nuove connessioni vengono aggiunte alla coda di backlog e verrà attesa la chiusura delle altre connessioni. Se il parametro è impostato su 1, ElastiCache invia un errore al client e chiude immediatamente la connessione.</p>

Nome	Descrizione
s l ab_automove	<p>Regola l'algoritmo slab automove: se questo parametro è impostato su 0 (zero), l'algoritmo è disabilitato. Se è impostato su 1, ElastiCache utilizza un lento approccio tradizionale per lo spostamento automatico degli slab. Se è impostato su 2, ElastiCache sposta improvvisamente gli slab ogni volta che si verifica un'espulsione. Questa modalità è consigliabile esclusivamente a scopo di test.</p>

Nome	Descrizione
<code>slab_reassign</code>	Abilitare o disabilitare la riassegnazione dello slab. Se questo parametro è impostato su 1, puoi utilizzare il comando di "riassegnazione slab" (<code>slab_reassign</code>) per riassegnare manualmente la memoria.

Parametri di Memcached 1.4.5 supportati

Famiglia del gruppo di parametri: `memcached1.4`

Per Memcached 1.4.5, sono supportati i seguenti parametri.

Parametri aggiunti in Memcached 1.4.5

Nome	Informazioni	Descrizione
<code>backlog_queue_limit</code>	Impostazione predefinita: 1024 Tipo: Integer Modificabile: no	Limite della coda di backlog.

Nome	Informazioni	Descrizione
<code>binding_protocol</code>	<p>Impostazione predefinita: <code>auto</code></p> <p>Tipo: stringa</p> <p>Modificabile: sì</p> <p>Le modifiche hanno effetto: Dopo il riavvio</p>	<p>Il protocollo di binding.</p> <p>Valori consentiti: <code>ascii</code> e <code>auto</code>.</p> <p>Per ulteriori informazioni sulla modifica del valore di <code>binding_protocol</code>, consulta Modifica di un gruppo di parametri.</p>
<code>cas_disabled</code>	<p>Impostazione predefinita: <code>0(false)</code></p> <p>Tipo: Booleano</p> <p>Modificabile: sì</p> <p>Le modifiche hanno effetto: Dopo il riavvio</p>	<p>Se <code>1 (true)</code>, le operazioni di controllo e impostazione (CAS) verranno disabilitate e gli elementi archiviati utilizzeranno 8 byte in meno rispetto a quando CAS è abilitato.</p>
<code>chunk_size</code>	<p>Impostazione predefinita: <code>48</code></p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche hanno effetto: Dopo il riavvio</p>	<p>La quantità minima, in byte, di spazio da allocare per chiave, valore e flag dell'elemento più piccolo.</p>
<code>chunk_size_growth_factor</code>	<p>Impostazione predefinita: <code>1.25</code></p> <p>Tipo: Float</p> <p>Modificabile: sì</p> <p>Le modifiche hanno effetto: Dopo il riavvio</p>	<p>Il fattore di crescita che controlla le dimensioni di ciascun successivo blocco di Memcached. Ciascun blocco sarà maggiore del precedente e in base al valore di <code>chunk_size_growth_factor</code>.</p>

Nome	Informazioni	Descrizione
<code>error_on_memory_exhausted</code>	Impostazione predefinita: 0(false) Tipo: Booleano Modificabile: sì Le modifiche hanno effetto: Dopo il riavvio	Se 1 (true), quando si esaurisce la memoria per archiviare gli elementi, Memcached restituirà un errore anziché espellere gli elementi.
<code>large_memory_pages</code>	Impostazione predefinita: 0(false) Tipo: Booleano Modificabile: no	Se 1 (true), ElastiCache tenterà di utilizzare pagine di memoria di grandi dimensioni.
<code>lock_down_paged_memory</code>	Impostazione predefinita: 0(false) Tipo: Booleano Modificabile: no	Se 1 (true), ElastiCache bloccherà tutta la memoria paginata.
<code>max_item_size</code>	Impostazione predefinita: 1048576 Tipo: Integer Modificabile: sì Le modifiche hanno effetto: Dopo il riavvio	La dimensione, in byte, dell'elemento più grande che può essere archiviato nel cluster.

Nome	Informazioni	Descrizione
<code>max_simultaneous_connections</code>	Impostazione predefinita: 65000 Tipo: Integer Modificabile: no	Il numero massimo di connessioni simultanee.
<code>maximize_core_file_limit</code>	Impostazione predefinita: 0(false) Tipo: Booleano Modificabile: Le modifiche hanno effetto: Dopo il riavvio	Se 1 (true), ElastiCache aumenterà il limite del file core.
<code>memcached_connections_overhead</code>	Impostazione predefinita: 100 Tipo: Integer Modificabile: sì Le modifiche hanno effetto: Dopo il riavvio	La quantità di memoria da riservare per le connessioni Memcached e altro sovraccarico vario. Per informazioni su questo parametro , consulta Sovraccarico delle connessioni Memcached .
<code>requests_per_event</code>	Impostazione predefinita: 20 Tipo: Integer Modificabile: no	Il numero massimo di richieste per evento per una connessione specifica. Questo limite è necessario per evitare un uso eccessivo delle risorse.

Sovraccarico delle connessioni Memcached

Su ciascun nodo, la memoria resa disponibile per l'archiviazione degli elementi è la memoria totale disponibile sul nodo, archiviata nel parametro `max_cache_memory`, meno la memoria utilizzata per le connessioni e altro sovraccarico, archiviata nel parametro `memcached_connections_overhead`. Un nodo di tipo `cache.m1.small`

ha ad esempio un valore di `max_cache_memory` pari a 1300 MB. Con il valore `memcached_connections_overhead` di default di 100 MB, il processo Memcached avrà 1200 MB disponibili per archiviare gli elementi.

I valori predefiniti per il parametro `memcached_connections_overhead` sono sufficienti nella maggior parte dei casi. La quantità di allocazione richiesta per il sovraccarico delle connessioni può tuttavia variare in base a più fattori, tra cui tasso di richiesta, dimensione del payload e numero di connessioni.

Puoi modificare il valore di `memcached_connections_overhead` per soddisfare al meglio le esigenze della tua applicazione. Aumentando ad esempio il valore del parametro `memcached_connections_overhead` otterrai la riduzione della quantità di memoria disponibile per l'archiviazione degli elementi e verrà reso disponibile un buffer di dimensioni maggiori per il sovraccarico delle connessioni. Riducendo il valore del parametro `memcached_connections_overhead` avrai a disposizione una maggiore quantità di memoria per archiviare gli elementi, sebbene ciò aumenti il rischio di utilizzo di swap e di riduzione delle prestazioni. Se riscontri utilizzo dello swap e riduzione delle prestazioni, prova ad aumentare il valore del parametro `memcached_connections_overhead`.

Important

Per il tipo di nodo `cache.t1.micro`, il valore di `memcached_connections_overhead` viene determinato come segue:

- Se il cluster utilizza il gruppo di parametri di default, ElastiCache imposterà il valore di `memcached_connections_overhead` su 13 MB.
- Se il cluster utilizza un gruppo di parametri che tu hai creato, puoi impostare `memcached_connections_overhead` su un valore di tua scelta.

Parametri specifici del tipo di nodo Memcached

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 i valori predefiniti dei parametri `max_cache_memory` e `num_threads` per ciascun tipo di nodo. I valori di questi parametri non possono essere modificati.

Tipo di nodo	max_cache_memory (in megabyte)	num_threads
cache.t1.micro	213	1
cache.t2.micro	555	1
cache.t2.small	1588	1
cache.t2.medium	3301	2
cache.t3.micro	512	2
cache.t3.small	1402	2
cache.t3.medium	3364	2
cache.t4g.micro	512	2
cache.t4g.small	1402	2
cache.t4g.medium	3164	2
cache.m1.small	1301	1
cache.m1.medium	3350	1
cache.m1.large	7100	2
cache.m1.xlarge	14600	4
cache.m2.xlarge	33800	2
cache.m2.2xlarge	30412	4
cache.m2,4xlarge	68000	16
cache.m3.medium	2850	1
cache.m3.large	6200	2
cache.m3.xlarge	13600	4

Tipo di nodo	max_cache_memory (in megabyte)	num_threads
cache.m3,2xlarge	28600	8
cache.m4.large	6573	2
cache.m4.xlarge	11496	4
cache.m4.2xlarge	30412	8
cache.m4.4xlarge	62234	16
cache.m4.10xlarge	158355	40
cache.m5.large	6537	2
cache.m5.xlarge	13248	4
cache.m5.2xlarge	26671	8
cache.m5.4xlarge	53516	16
cache.m5.12xlarge	160900	48
cache.m5.24xlarge	321865	96
cache.m6g.large	6537	2
cache.m6g.xlarge	13248	4
cache.m6g.2xlarge	26671	8
cache.m6g.4xlarge	53516	16
cache.m6g.8xlarge	107000	32
cache.m6g.12xlarge	160900	48
cache.m6g.16xlarge	214577	64
cache.c1.xlarge	6600	8

Tipo di nodo	max_cache_memory (in megabyte)	num_threads
cache.r3.large	13800	2
cache.r3.xlarge	29100	4
cache.r3.2xlarge	59600	8
cache.r3.4xlarge	120600	16
cache.r3.8xlarge	120600	32
cache.r4.large	12590	2
cache.r4.xlarge	25652	4
cache.r4.2xlarge	51686	8
cache.r4.4xlarge	103815	16
cache.r4.8xlarge	208144	32
cache.r4.16xlarge	416776	64
cache.r5.large	13387	2
cache.r5.xlarge	26953	4
cache.r5.2xlarge	54084	8
cache.r5.4xlarge	108347	16
cache.r5.12xlarge	325400	48
cache.r5.24xlarge	650869	96
cache.r6g.large	13387	2
cache.r6g.xlarge	26953	4
cache.r6g.2xlarge	54084	8

Tipo di nodo	max_cache_memory (in megabyte)	num_threads
cache.r6g.4xlarge	108347	16
cache.r6g.8xlarge	214577	32
cache.r6g.12xlarge	325400	48
cache.r6g.16xlarge	429154	64
cache.c7gn.large	3164	2
cache.c7gn.xlarge	6537	4
cache.c7gn.2xlarge	13248	8
cache.c7gn.4xlarge	26671	16
cache.c7gn.8xlarge	53516	32
cache.c7gn.12xlarge	325400	48
cache.c7gn.16xlarge	108347	64

Note

Tutte le istanze T2 vengono create in Amazon Virtual Private Cloud (Amazon VPC).

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](#)
- [Modifiche dei parametri di Redis 6.x](#)
- [Modifiche dei parametri di Redis 5.0.3](#)
- [Modifiche dei parametri di Redis 5.0.0](#)
- [Modifiche dei parametri di Redis 4.0.10](#)
- [Modifiche dei parametri di Redis 3.2.10](#)
- [Modifiche dei parametri di Redis 3.2.6](#)
- [Modifiche dei parametri di Redis 3.2.4](#)
- [Parametri aggiunti in Redis 2.8.24 \(avanzato\)](#)
- [Parametri aggiunti in Redis 2.8.23 \(avanzato\)](#)
- [Parametri aggiunti in Redis 2.8.22 \(avanzato\)](#)
- [Parametri aggiunti in Redis 2.8.21](#)
- [Parametri aggiunti in Redis 2.8.19](#)
- [Parametri aggiunti in Redis 2.8.6](#)
- [Parametri di Redis 2.6.13](#)
- [Parametri specifici del tipo di nodo di Redis](#)

Modifiche dei parametri di Redis 7

Famiglia del gruppo di parametri: `redis7`

I gruppi di parametri predefiniti di Redis 7 sono i seguenti:

- `default.redis7`— Utilizzare questo gruppo di parametri, o uno derivato da esso, per i cluster Redis (modalità cluster disabilitata) e i gruppi di replica.

- `default.redis7.cluster.on`— Utilizzare questo gruppo di parametri, o uno derivato da esso, per i cluster Redis (modalità cluster abilitata) e i gruppi di replica.

I parametri aggiunti in Redis 7 sono i seguenti.

Nome	Informazioni	Descrizione
<code>cluster-allow-pubsubshard-when-down</code>	<p>Valori consentiti: <code>yes</code>, <code>no</code></p> <p>Impostazione predefinita: <code>yes</code></p> <p>Tipo: <code>string</code></p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente in tutti i nodi del cluster.</p>	<p>Se impostato sul valore predefinito <code>yes</code>, consente ai nodi di servire il traffico di partizion e pubsub mentre il cluster è in uno stato inattivo, purché ritenga di possedere gli slot.</p>
<code>cluster-preferred-endpoint-type</code>	<p>Valori consentiti: <code>ip</code>, <code>tls-dynamic</code></p> <p>Impostazione predefinita: <code>tls-dynamic</code></p> <p>Tipo: <code>string</code></p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente in tutti i nodi del cluster.</p>	<p>Questo valore controlla quale endpoint viene restituito per le richieste <code>MOVED/ASKING</code> nonché il campo <code>endpoint</code> per <code>CLUSTER SLOTS</code> e <code>CLUSTER SHARDS</code>. Quando il valore è impostato su <code>ip</code>, il nodo pubblicizzerà il suo indirizzo IP. Quando il valore è impostato su <code>tls-dynamic</code>, il nodo segnalerà un nome host quando la crittografia dei dati in transito è abilitata e un indirizzo IP in caso contrario.</p>
<code>latency-tracking</code>	<p>Valori consentiti: <code>yes</code>, <code>no</code></p> <p>Impostazione predefinita: <code>no</code></p> <p>Tipo: <code>string</code></p>	<p>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</p>

Nome	Informazioni	Descrizione
	<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.
<code>list-max-ziplist-size</code>	<p>Valori consentiti:</p> <p>Impostazione predefinita: -2</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente in tutti i nodi del cluster.</p>	Il numero di voci consentite per nodo elenco interno.

Modifiche dei parametri di Redis 6.x

Famiglia del gruppo di parametri: `redis6.x`

I gruppi di parametri predefiniti di Redis 6.x sono i seguenti:

- `default.redis6.x`— Utilizzare questo gruppo di parametri, o uno derivato da esso, per i cluster Redis (modalità cluster disabilitata) e i gruppi di replica.
- `default.redis6.x.cluster.on`— Utilizzare questo gruppo di parametri, o uno derivato da esso, per i cluster Redis (modalità cluster abilitata) e i gruppi di replica.

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 policy noeviction, volatile-lru e allkeys-lru max-memory sono supportate con i tipi di nodi r6gd.

Per ulteriori informazioni, consulta [ElastiCache for Redis versione 6.2 \(avanzato\)](#) e [ElastiCache for Redis versione 6.0 \(avanzata\)](#).

I parametri aggiunti in Redis 6.x sono i seguenti.

Nome	Informazioni	Descrizione
<code>acl-pubsub-default</code> (added in 6.2)	<p>Valori consentiti: <code>resetchannels</code>, <code>allchannels</code></p> <p>Impostazione predefinita: <code>allchannels</code></p> <p>Tipo: string</p> <p>Modificabile: sì</p> <p>Le modifiche entrano in vigore: gli utenti di Redis esistenti associati al cluster continueranno a disporre delle autorizzazioni esistenti. Aggiorna gli utenti o riavvia il cluster per aggiornare gli utenti Redis esistenti.</p>	Autorizzazioni di canale pubsub di default per gli utenti ACL implementati in questo cluster.
<code>cluster-allow-reads-when-down</code> (added in 6.0)	<p>Impostazione predefinita: no</p> <p>Tipo: stringa</p> <p>Modificabile: sì</p>	Se impostato su sì, un gruppo di replica Redis (modalità cluster abilitata) continua a elaborare i comandi di lettura anche quando un nodo non è in grado di raggiungere un quorum di primarie.

Nome	Informazioni	Descrizione
	Le modifiche diventano effettive immediatamente in tutti i nodi del cluster	Se impostato sul valore di default su no, il gruppo di replica rifiuta tutti i comandi. Si consiglia di impostare questo valore su sì se si utilizza un cluster con meno di tre gruppi di nodi o se l'applicazione è in grado di gestire in modo sicuro le letture non aggiornate.
tracking-table-max-keys (added in 6.0)	<p>Impostazione di default: 1.000.000</p> <p>Tipo: numero</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive immediatamente in tutti i nodi del cluster</p>	<p>Per aiutare il caching lato client, Redis supporta il monitoraggio di quali client hanno accesso a quali chiavi.</p> <p>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. Dopo aver superato questo valore di parametro, i client vengono inviati invalidazione casualmente. Questo valore dovrebbe essere sintonizzato per limitare l'utilizzo della memoria pur mantenendo traccia di un numero sufficiente di chiavi. Anche le chiavi vengono invalidate in condizioni di memoria insufficiente.</p>
acllog-max-len (added in 6.0)	<p>Impostazione predefinita: 128</p> <p>Tipo: numero</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive immediatamente in tutti i nodi del cluster</p>	Questo valore corrisponde al numero massimo di voci nel registro ACL.

Nome	Informazioni	Descrizione
<code>active-expire-effort</code> (added in 6.0)	<p>Impostazione predefinita: 1</p> <p>Tipo: numero</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive immediatamente in tutti i nodi del cluster</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> (added in 6.0)	<p>Impostazione predefinita: no</p> <p>Tipo: stringa</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive immediatamente in tutti i nodi del cluster</p>	<p>Quando il valore è impostato su yes, la proprietà DEL agisce come UNLINK.</p>

I parametri rimossi in Redis 6.x sono i seguenti.

Nome	Informazioni	Descrizione
<code>lua-replicate-commands</code>	<p>Valori consentiti: <code>yes/no (sì/no)</code></p> <p>Impostazione predefinita: <code>yes (sì)</code></p> <p>Tipo: booleano</p> <p>Modificabile: <code>sì</code></p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Abilitare/disabilitare sempre la replica effetti Lua negli script Lua.</p>

Modifiche dei parametri di Redis 5.0.3

Famiglia del gruppo di parametri: `redis5.0`

Gruppi di parametri predefiniti di Redis 5.0

- `default.redis5.0`— Utilizzare questo gruppo di parametri, o uno derivato da esso, per i cluster Redis (modalità cluster disabilitata) e i gruppi di replica.
- `default.redis5.0.cluster.on`— Utilizzare questo gruppo di parametri, o uno derivato da esso, per i cluster Redis (modalità cluster abilitata) e i gruppi di replica.

Parametri aggiunti in Redis 5.0.3

Nome	Informazioni	Descrizione
<code>rename-commands</code>	<p>Impostazione predefinita: <code>none</code></p> <p>Tipo: stringa</p> <p>Modificabile: <code>sì</code></p> <p>Le modifiche diventano effettive immediatamente in tutti i nodi del cluster</p>	<p>Un elenco separato da spazi di comandi Redis rinominati. Di seguito è riportato un elenco limitato di comandi disponibili per la ridenominazione:</p> <p><code>APPEND AUTH BITCOUNT BITFIELD BITOP BITPOS BLPOP BRPOP BRPOPLUSH BZPOPMIN BZPOPMAX CLIENT CLUSTER COMMAND DBSIZE DECR</code></p>

Nome	Informazioni	Descrizione
		<p> DECRBY DEL DISCARD DUMP ECHO EVAL EVALSHA EXEC EXISTS EXPIRE EXPIREAT FLUSHALL FLUSHDB GEOADD GEOHASH GEOPOS GEODIST GEORADIUS GEORADIUSBYMEMBER GET GETBIT GETRANGE GETSET HDEL HEXISTS HGET HGETALL HINCRBY HINCRBYFL OAT HKEYS HLEN HMGET HMSET HSET HSETNX HSTRLEN HVALS INCR INCRBY INCRBYFLOAT INFO KEYS LASTSAVE LINDEX LINSERT LLEN LPOP LPU SH LPUSHX LRANGE LREM LSET LTRIM MEMORY MGET MONITOR MOVE MSET MSETNX MULTI OBJECT PERSIST PEXPIRE PEXPIREAT PFADD PFCOUNT PFMERGE PING PSETEX PSUBSCRIBE PUBSUB PTTL PUBLISH PUNSUBSCRIBE RANDOMKEY READONLY READWRITE RENAME RENAMENX RESTORE ROLE RPOP RPOPLPUSH RPUSH RPUSHX SADD SCARD SCRIPT SDIFF SDIFFSTORE SELECT SET SETBIT SETEX SETNX SETRANGE SINTER SINTERSTORE SISMEMBER SLOWLOG SMEMBERS SMOVE SORT SPOP SRANDMEMBER SREM STRLEN SUBSCRIBE UNION UNIONSTORE SWAPDB TIME TOUCH TTL TYPE UNSUBSCRIBE UNLINK UNWATCH WAIT WATCH ZADD ZCARD ZCOUNT ZINCRBY ZINTERSTO RE ZLEXCOUNT ZPOPMAX ZPOPMIN ZRANGE ZRANGEBYLEX ZREVRANGE BYLEX ZRANGEBYSCORE ZRANK ZREM ZREMRANGEBYLEX ZREMRANGEBYRANK ZREMRANGEBYSCORE ZREVRANGE </p>

Nome	Informazioni	Descrizione
		ZREVRANGEBYSCORE ZREVRANK ZSCORE ZUNIONSTORE SCAN SSCAN HSCAN ZSCAN XINFO XADD XTRIM XDEL XRA NGE XREVRANGE XLEN XREAD XGROUP XREADGROUP XACK XCLAIM XPENDING GEORADIUS_RO GEORADIUSBYMEMBER_ RO LOLWUT XSETID SUBSTR

Per ulteriori informazioni, consulta [ElastiCache for Redis versione 5.0.6 \(potenziata\)](#).

Modifiche dei parametri di Redis 5.0.0

Famiglia del gruppo di parametri: redis5.0

Gruppi di parametri predefiniti di Redis 5.0

- `default.redis5.0`— Utilizzare questo gruppo di parametri, o uno derivato da esso, per i cluster Redis (modalità cluster disabilitata) e i gruppi di replica.
- `default.redis5.0.cluster.on`— Utilizzare questo gruppo di parametri, o uno derivato da esso, per i cluster Redis (modalità cluster abilitata) e i gruppi di replica.

Parametri aggiunti in Redis 5.0

Nome	Informazioni	Descrizione
<code>stream-node-max-bytes</code>	Valori consentiti: 0+ Impostazione predefinita: 4096 Tipo: Integer Modificabile: sì Le modifiche diventano effettive: immediatamente	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.

Nome	Informazioni	Descrizione
<code>stream-node-max-entries</code>	<p>Valori consentiti: 0+</p> <p>Impostazione predefinita: 100</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>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 impostata su 0, il numero delle voci nel nodo della struttura è illimitato</p>
<code>active-defrag-max-scan-fields</code>	<p>Valori consentiti: da 1 a 1000000</p> <p>Impostazione predefinita: 1000</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Numero massimo di campi set/hash/zset/list che saranno elaborati dalla scansione del dizionario principale</p>
<code>lua-replicate-commands</code>	<p>Valori consentiti: yes/no (sì/no)</p> <p>Impostazione predefinita: yes (sì)</p> <p>Tipo: booleano</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Abilitare/disabilitare sempre la replica effetti Lua negli script Lua.</p>

Nome	Informazioni	Descrizione
<code>replica-ignore-maxmemory</code>	Impostazione predefinita: yes (sì) Tipo: booleano Modificabile: no	Determina se la replica ignora l'impostazione <code>maxmemory</code> senza espellere gli elementi indipendenti dal primario

Redis ha ridenominato numerosi parametri nel motore versione 5.0 in risposta al feedback della community. Per ulteriori informazioni, consulta la sezione relativa alle [novità di Redis 5](#). La tabella seguente elenca i nuovi nomi e il modo in cui vengono mappati alle versioni precedenti.

Parametri rinominati in Redis 5.0

Nome	Informazioni	Descrizione
<code>replica-lazy-flush</code>	Impostazione predefinita: yes (sì) Tipo: booleano Modificabile: no Nome precedente: <code>slave-lazy-flush</code>	Esegue un'operazione <code>flushDB</code> asincrona durante la sincronizzazione della replica.
<code>client-output-buffer-limit-replica-hard-limit</code>	Valore di default: per i valori, vedere Parametri specifici del tipo di nodo di Redis Tipo: Integer Modificabile: no Nome precedente: <code>client-output-buffer-limit-slave-hard-limit</code>	Per le repliche di lettura Redis: se il buffer di output di un client raggiunge il numero di byte specificato, il client verrà disconnesso.

Nome	Informazioni	Descrizione
<code>client-output-buffer-limit-replica-soft-limit</code>	<p>Valore di default: per i valori, vedere Parametri specifici del tipo di nodo di Redis</p> <p>Tipo: Integer</p> <p>Modificabile: no</p> <p>Nome precedente: <code>client-output-buffer-limit-slave-soft-limit</code></p>	Per le repliche di lettura Redis: 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-replica-soft-seconds</code> .
<code>client-output-buffer-limit-replica-soft-seconds</code>	<p>Impostazione predefinita: 60</p> <p>Tipo: Integer</p> <p>Modificabile: no</p> <p>Nome precedente: <code>client-output-buffer-limit-slave-soft-seconds</code></p>	Per le repliche di lettura Redis: se il buffer di output di un client rimane di <code>client-output-buffer-limit-replica-soft-limit</code> byte per un tempo maggiore del numero di secondi specificato, il client verrà disconnesso.
<code>replica-allow-chaining</code>	<p>Impostazione predefinita: no</p> <p>Tipo: stringa</p> <p>Modificabile: no</p> <p>Nome precedente: <code>slave-allow-chaining</code></p>	Determina se una replica di lettura in Redis può disporre di repliche di lettura specifiche.

Nome	Informazioni	Descrizione
<code>min-replicas-to-write</code>	<p>Impostazione Predefinita: 0</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Nome precedente: <code>min-slaves-to-write</code></p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Il numero minimo di repliche di lettura che deve essere disponibile affinché il nodo principale accetti le operazioni di scrittura dai client. Se il numero di repliche disponibili scende al di sotto di questo numero, il nodo principale non accetterà più richieste di scrittura.</p> <p>Se questo parametro o <code>min-replicas-max-lag</code> è 0, il nodo principale accetterà sempre le richieste di scrittura, anche se non sono disponibili repliche.</p>
<code>min-replicas-max-lag</code>	<p>Impostazione predefinita: 10</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Nome precedente: <code>min-slaves-max-lag</code></p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Il numero di secondi entro i quali il nodo principale deve ricevere una richiesta di ping da una replica di lettura. Se trascorre questa quantità di tempo e il nodo principale non riceve un ping, la replica non viene più considerata disponibile. Se il numero di repliche disponibili scende al di sotto del valore di <code>min-replicas-to-write</code>, il nodo principale non accetta più operazioni di scrittura.</p> <p>Se questo parametro o <code>min-replicas-to-write</code> è 0, il nodo principale accetterà sempre le richieste di scrittura, anche se non sono disponibili repliche.</p>

Nome	Informazioni	Descrizione
<code>close-on-replica-write</code>	Impostazione predefinita: <code>yes</code> (sì) Tipo: booleano Modificabile: sì Nome precedente: <code>close-on-slave-write</code> Le modifiche diventano effettive: immediatamente	Se abilitato, i client che tentano di scrivere in una replica di sola lettura verranno disconnessi.

Parametri rimossi in Redis 5.0

Nome	Informazioni	Descrizione
<code>repl-timeout</code>	Impostazione predefinita: <code>60</code> Modificabile: no	Il parametro non è disponibile in questa versione.

Modifiche dei parametri di Redis 4.0.10

Famiglia del gruppo di parametri: `redis4.0`

Gruppi di parametri predefiniti di Redis 4.0

- `default.redis4.0` - Utilizzare questo gruppo di parametri, o uno derivato da esso, per i cluster Redis (modalità cluster disabilitata) e i gruppi di replica.
- `default.redis4.0.cluster.on` - Utilizzare questo gruppo di parametri, o uno derivato da esso, per i cluster Redis (modalità cluster abilitata) e i gruppi di replica.

Parametri modificati in Redis 4.0.10

Nome	Informazioni	Descrizione
maxmemory-policy	<p>Valori consentiti: allkeys-lru , volatile-lru , allkeys-lfu , volatile-lfu , allkeys-random , volatile-random , volatile-ttl , noeviction</p> <p>Impostazione predefinita: volatile-lru</p> <p>Tipo: stringa</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente.</p>	<p>maxmemory-policy è stato aggiunto nella versione 2.6.13. Nella versione 4.0.10 sono stati aggiunti due nuovi valori ammessi: allkeys-lfu , che eliminerà qualsiasi chiave utilizzando l'LFU approssimata e volatile-lfu , che effettuerà le eliminazioni utilizzando l'LFU approssimata tra le chiavi con una scadenza impostata. Nella versione 6.2, quando la famiglia di nodi r6gd è stata introdotta per l'uso con il tiering di dati, solo le policy noeviction , volatile-lru e allkeys-lru max-memory sono supportate con i tipi di nodi r6gd.</p>

Parametri aggiunti in Redis 4.0.10

Nome	Informazioni	Descrizione
Parametri di eliminazione asincrona		
lazyfree-lazy- eviction	<p>Valori consentiti: yes/no (sì/no)</p> <p>Impostazione predefinita: no</p> <p>Tipo: booleano</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente.</p>	<p>Performs an asynchronous delete on evictions.</p>

Nome	Informazioni	Descrizione
<code>lazyfree-lazy-expire</code>	Valori consentiti: yes/no (sì/no) Impostazione predefinita: no Tipo: booleano Modificabile: sì Le modifiche diventano effettive: immediatamente.	Performs an asynchronous delete on expired keys.
<code>lazyfree-lazy-server-del</code>	Valori consentiti: yes/no (sì/no) Impostazione predefinita: no Tipo: booleano Modificabile: sì Le modifiche diventano effettive: immediatamente.	Performs an asynchronous delete for commands which update values.
<code>slave-lazy-flush</code>	Valori consentiti: N/D Impostazione predefinita: no Tipo: booleano Modificabile: no Changes take place: N/A	Performs an asynchronous flushDB during slave sync.

Parametri LFU

Nome	Informazioni	Descrizione
<code>lfu-log-factor</code>	<p>Valori consentiti: qualsiasi numero intero > 0</p> <p>Impostazione predefinita: 10</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente.</p>	<p>Set the log factor, which determines the number of key hits to saturate the key counter.</p>
<code>lfu-decadimento</code>	<p>Valori consentiti: qualsiasi numero intero</p> <p>Impostazione predefinita: 1</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente.</p>	<p>The amount of time in minutes to decrement the key counter.</p>
Parametri di deframmentazione attivi		
<code>activedefrag</code>	<p>Valori consentiti: yes/no (sì/no)</p> <p>Impostazione predefinita: no</p> <p>Tipo: booleano</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente.</p>	<p>Enabled active defragmentation.</p>

Nome	Informazioni	Descrizione
<code>active-defrag-ignore-byte</code>	<p>Valori consentiti: 10485760-104857600</p> <p>Impostazione predefinita: 104857600</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente.</p>	Minimum amount of fragmentation waste to start active defrag.
<code>active-defrag-threshold-lower</code>	<p>Valori consentiti: 1-100</p> <p>Impostazione predefinita: 10</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente.</p>	Minimum percentage of fragmentation to start active defrag.
<code>active-defrag-threshold-upper</code>	<p>Valori consentiti: 1-100</p> <p>Impostazione predefinita: 100</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente.</p>	Maximum percentage of fragmentation at which we use maximum effort.

Nome	Informazioni	Descrizione
<code>active-defrag-cycle-min</code>	Valori consentiti: 1-75 Impostazione predefinita: 25 Tipo: Integer Modificabile: sì Le modifiche diventano effettive: immediatamente.	Minimal effort for defrag in CPU percentage.
<code>active-defrag-cycle-max</code>	Valori consentiti: 1-75 Impostazione predefinita: 75 Tipo: Integer Modificabile: sì Le modifiche diventano effettive: immediatamente.	Maximal effort for defrag in CPU percentage.
Parametri buffer di output client		
<code>client-query-buffer-limit</code>	Valori consentiti: 1048576-1073741824 Impostazione predefinita: 1073741824 Tipo: Integer Modificabile: sì Le modifiche diventano effettive: immediatamente.	Max size of a single client query buffer.

Nome	Informazioni	Descrizione
<code>proto-max-bulk-len</code>	<p>Valori consentiti: 1048576-536870912</p> <p>Impostazione predefinita: 536870912</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente.</p>	Max size of a single element request.

Modifiche dei parametri di Redis 3.2.10

Famiglia del gruppo di parametri: `redis3.2`

ElastiCache for Redis 3.2.10 non sono supportati parametri aggiuntivi.

Modifiche dei parametri di Redis 3.2.6

Famiglia del gruppo di parametri: `redis3.2`

Per Redis 3.2.6 non sono supportati parametri aggiuntivi.

Modifiche dei parametri di Redis 3.2.4

Famiglia del gruppo di parametri: `redis3.2`

A partire da Redis 3.2.4 esistono due gruppi di parametri predefiniti.

- `default.redis3.2`— Quando si esegue Redis 3.2.4, specificare questo gruppo di parametri o uno derivato da esso, se si desidera creare un gruppo di replica Redis (modalità cluster disabilitata) e utilizzare comunque le caratteristiche aggiuntive di Redis 3.2.4.
- `default.redis3.2.cluster.on` - Specificare questo gruppo di parametri o uno derivato da esso, quando si vuole creare un gruppo di replica Redis (modalità cluster abilitata).

Argomenti

- [Nuovi parametri per Redis 3.2.4](#)
- [Parametri modificati in Redis 3.2.4 \(avanzato\)](#)

Nuovi parametri per Redis 3.2.4

Famiglia del gruppo di parametri: redis3.2

Per Redis 3.2.4, sono supportati i seguenti parametri aggiuntivi.

Nome	Informazioni	Descrizione
<code>list-max-ziplist-size</code>	Impostazione predefinita: -2 Tipo: Integer Modificabile: no	<p>Gli elenchi sono codificati in un modo speciale per risparmiare spazio. Il numero di voci consentite per il nodo elenco interno può essere specificato come dimensione massima fissa o numero massimo di elementi. Per una dimensione massima fissa, utilizza da -5 a -1, ovvero:</p> <ul style="list-style-type: none">• -5: dimensione massima: 64 KB (non consigliato per carichi di lavoro normali)• -4: dimensione massima: 32 KB (non consigliato)• -3: dimensione massima: 16 KB (non consigliato)• -2: dimensione massima: 8 KB (consigliato)• -1: dimensione massima: 4 KB (consigliato)• I numeri positivi indicano l'archiviazione esattamente fino al numero specificato di elementi per nodo elenco.

Nome	Informazioni	Descrizione
list-compress-depth	Impostazione Predefinita: 0 Tipo: Integer Modificabile: sì Le modifiche diventano effettive: immediatamente	<p>Gli elenchi possono anche essere compressi . 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>cluster-enabled</code>	<p>Impostazione predefinita: no/yes (no/sì) *</p> <p>Tipo: string</p> <p>Modificabile: no</p>	<p>Indica se si tratta di un gruppo di replica Redis (modalità cluster abilitata) in modalità cluster (sì) o di un gruppo di replica Redis (modalità cluster abilitata) in modalità non cluster (no). I gruppi di replica Redis (modalità cluster abilitata) in modalità cluster possono ripartire i propri dati tra un massimo di 500 gruppi di nodi.</p> <p>* Redis 3.2.x ha due gruppi di parametri predefiniti.</p> <ul style="list-style-type: none">• <code>default.redis3.2</code> – Valore di default no.• <code>default.redis3.2.cluster.on</code> – Valore di default yes. <p>.</p>

Nome	Informazioni	Descrizione
<code>cluster-require-full-coverage</code>	<p>Impostazione predefinita: no</p> <p>Tipo: booleano</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Se impostato su <code>yes</code>, i nodi Redis (modalità cluster abilitata) in modalità cluster smettono di accettare le query se rilevano che è presente almeno un hash slot non coperto (non gestito da alcun nodo disponibile). In tal caso, se il cluster è parzialmente inattivo, il cluster diventa non disponibile. Diventa automaticamente di nuovo disponibile non appena tutti gli slot sono nuovamente coperti.</p> <p>In alcuni casi, è possibile fare in modo che il sottoinsieme del cluster funzionante continui ad accettare le query da parte dello spazio della chiave ancora coperto. A tale scopo, imposta l'opzione <code>cluster-require-full-coverage</code> su <code>no</code>.</p>

Nome	Informazioni	Descrizione
hll-sparse-max-bytes	<p>Impostazione predefinita: 3000</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Limite in byte della rappresentazione di tipo sparse di HyperLogLog. Il limite include l'intestazione a 16 byte. Quando un elemento HyperLogLog che utilizza la rappresentazione sparse supera tale limite, viene convertito in una rappresentazione di tipo dense.</p> <p>Un valore maggiore di 16000 non è consigliato, perché a quel punto la rappresentazione dense è più efficiente in termini di memoria.</p> <p>È consigliabile un valore di circa 3000 per ottenere i vantaggi della codifica efficiente degli spazi, senza rallentare troppo PFADD, che è impostato su $O(N)$ con la codifica sparse. Il valore può essere portato a circa 10000 quando il problema non è rappresentato dalla CPU, ma dallo spazio, e il set di dati è composto da numerosi elementi HyperLogLog con cardinalità nell'intervallo 0 - 15000.</p>

Nome	Informazioni	Descrizione
<code>reserved-memory-percent</code>	<p>Impostazione predefinita: 25</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>La percentuale di memoria di un nodo prenotata per un utilizzo diverso dai dati. Per impostazione predefinita, l'impatto dei dati Redis aumenta finché non viene utilizzata tutta la memoria del nodo. In tal caso, le prestazioni del nodo subiranno una riduzione a causa di un eccessivo scambio di pagine di memoria. Tramite l'allocazione della memoria, puoi riservare parte della memoria disponibile per attività non Redis, per consentire la riduzione della quantità di scambio di pagine.</p> <p>Questo parametro è specifico di ElastiCache e non fa parte della distribuzione Redis standard.</p> <p>Per ulteriori informazioni, consulta <code>reserved-memory</code> e Gestione della memoria prenotata.</p>

Parametri modificati in Redis 3.2.4 (avanzato)

Famiglia del gruppo di parametri: `redis3.2`

Per Redis 3.2.4 sono stati modificati i seguenti parametri.

Nome	Informazioni	Modifica
<code>activeresharding</code>	Modificabile: sì se il gruppo di parametri non è associato a un cluster di cache. In caso contrario, no.	Non era modificabile
<code>databases</code>	Modificabile: sì se il gruppo di parametri non è associato a un cluster di cache. In caso contrario, no.	Non era modificabile

Nome	Informazioni	Modifica
<code>appendonly</code>	Impostazione predefinita: off Modificabile: no	Se desideri eseguire l'aggiornamento da una versione precedente di Redis, devi prima disattivare <code>appendonly</code> .
<code>appendfsync</code>	Impostazione predefinita: off Modificabile: no	Se desideri eseguire l'aggiornamento da una versione precedente di Redis, devi prima disattivare <code>appendfsync</code> .
<code>repl-timeout</code>	Impostazione predefinita: 60 Modificabile: no	Ora non è modificabile con un valore di default di 60.
<code>tcp-keepalive</code>	Impostazione predefinita: 300	Il valore di default era 0.
<code>list-max-ziplist-entries</code>		Il parametro non è più disponibile.
<code>list-max-ziplist-value</code>		Il parametro non è più disponibile.

Parametri aggiunti in Redis 2.8.24 (avanzato)

Famiglia del gruppo di parametri: `redis2.8`

Per Redis 2.8.24 non sono supportati parametri aggiuntivi.

Parametri aggiunti in Redis 2.8.23 (avanzato)

Famiglia del gruppo di parametri: `redis2.8`

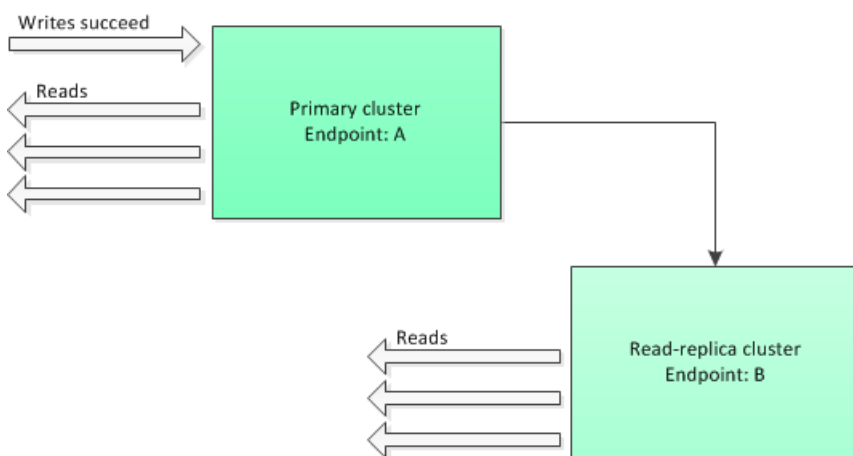
Per Redis 2.8.23, è supportato il seguente parametro aggiuntivo.

Nome	Informazioni	Descrizione
<code>close-on-slave-write</code>	Impostazione predefinita: <code>yes</code> (sì) Tipo: stringa (yes/no) Modificabile: sì Le modifiche diventano effettive: immediatamente	Se abilitato, i client che tentano di scrivere in una replica di sola lettura verranno disconnessi.

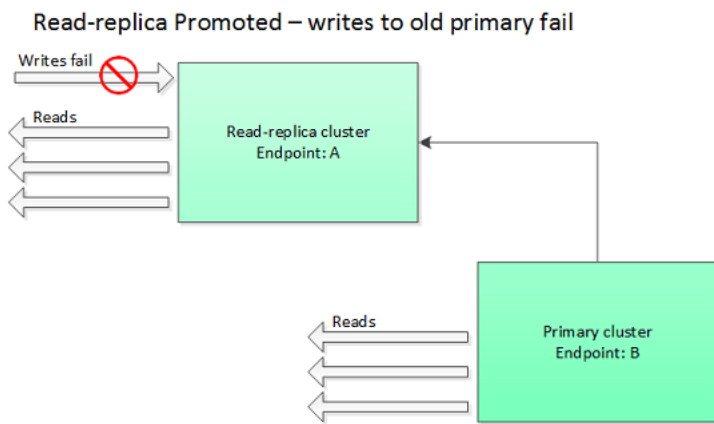
Come funziona `close-on-slave-write`

Il parametro `close-on-slave-write` viene introdotto in Amazon ElastiCache per fornirti maggiore controllo sulla modalità di risposta del cluster quando un nodo primario e un nodo di replica di lettura deviano i ruoli a causa della promozione di una replica di lettura in replica principale.

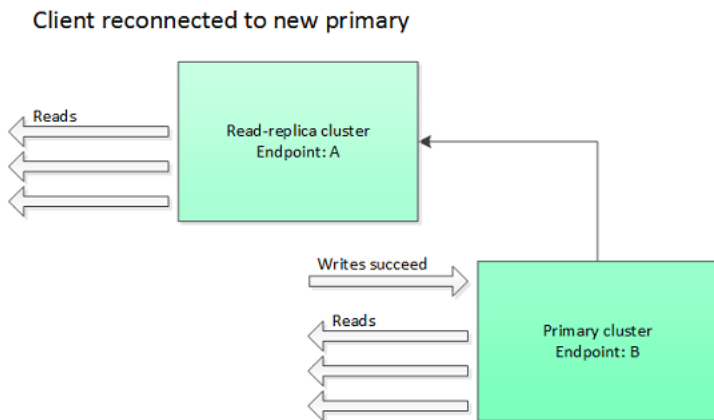
Before read-replica promotion



Se la replica di lettura al cluster viene promossa a replica principale per un motivo diverso dal failover del gruppo di replica con opzione Multi-AZ abilitata, il client continuerà a tentare di scrivere sull'endpoint A. Poiché l'endpoint A è ora l'endpoint per una replica di lettura, le operazioni di lettura avranno esito negativo. Si tratta del comportamento di Redis prima che in `close-on-replica-write` venisse introdotto ElastiCache e se si disabilita `close-on-replica-write`.



Con `close-on-replica-write` abilitato, ogni volta che un client tenta di scrivere in una replica di lettura, la connessione client al cluster viene chiusa. La logica dell'applicazione dovrebbe rilevare la disconnessione, controllare la tabella DNS ed eseguire la riconnessione all'endpoint principale, che ora è l'endpoint B.



Quando potresti disabilitare `close-on-replica-write`?

Se disabilitando `close-on-replica-write` le operazioni di scrittura vengono eseguite sul cluster in stato di errore, per quale motivo si dovrebbe eseguire la disabilitazione di `close-on-replica-write`?


Come indicato in precedenza, con `close-on-replica-write` abilitato, ogni volta che un client tenta di scrivere in una replica di lettura, la connessione client al cluster viene chiusa. Stabilire una nuova connessione verso i nodi richiede tempo. Pertanto, la disconnessione e la riconnessione come risultato di una richiesta di scrittura alla replica influiscono anche sulla latenza delle richieste di lettura che vengono soddisfatte attraverso la stessa connessione. Tale effetto persiste fino a quando non viene stabilita una nuova connessione. Se l'applicazione esegue un numero elevato di operazioni

di lettura o è molto sensibile alla latenza, potresti preferire che i tuoi client rimangano connessi per evitare una riduzione delle prestazioni in lettura.

Parametri aggiunti in Redis 2.8.22 (avanzato)

Famiglia del gruppo di parametri: redis2.8

Per Redis 2.8.22 non sono supportati parametri aggiuntivi.

 Important

- A partire da Redis versione 2.8.22, `repl-backlog-size` si applica al cluster principale e ai cluster di replica.
- A partire da Redis versione 2.8.22, il parametro `repl-timeout` non è supportato. Se viene modificato, ElastiCache esegue la sovrascrittura con il valore di default (60), come con `appendonly`.

I parametri seguenti non sono più supportati.

- `appendonly`
- `appendfsync`
- `repl-timeout`

Parametri aggiunti in Redis 2.8.21

Famiglia del gruppo di parametri: redis2.8

Per Redis 2.8.21 non sono supportati parametri aggiuntivi.

Parametri aggiunti in Redis 2.8.19

Famiglia del gruppo di parametri: redis2.8

Per Redis 2.8.19 non sono supportati parametri aggiuntivi.

Parametri aggiunti in Redis 2.8.6


Famiglia del gruppo di parametri: redis2.8

Per Redis 2.8.6, sono supportati i seguenti parametri aggiuntivi.

Nome	Informazioni	Descrizione
<code>min-slaves-max-lag</code>	<p>Impostazione predefinita: 10</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Il numero di secondi entro i quali il nodo principale deve ricevere una richiesta di ping da una replica di lettura. Se trascorre questa quantità di tempo e il nodo principale non riceve un ping, la replica non viene più considerata disponibile. Se il numero di repliche disponibili scende al di sotto del valore di <code>min-slaves-to-write</code>, il nodo principale non accetta più operazioni di scrittura.</p> <p>Se questo parametro o <code>min-slaves-to-write</code> è 0, il nodo principale accetterà sempre le richieste di scrittura, anche se non sono disponibili repliche.</p>
<code>min-slaves-to-write</code>	<p>Impostazione Predefinita: 0</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Il numero minimo di repliche di lettura che deve essere disponibile affinché il nodo principale accetti le operazioni di scrittura dai client. Se il numero di repliche disponibili scende al di sotto di questo numero, il nodo principale non accetterà più richieste di scrittura.</p> <p>Se questo parametro o <code>min-slaves-max-lag</code> è 0, il nodo principale accetterà sempre le richieste di scrittura, anche se non sono disponibili repliche.</p>

Nome	Informazioni	Descrizione
notify-keyspace-events	<p>Impostazione predefinita: (una stringa vuota)</p> <p>Tipo: stringa</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>I tipi di eventi keyspace di cui Redis può inviare una notifica ai client. Ogni tipo di evento è rappresentato da una lettera singola:</p> <ul style="list-style-type: none"> • K— Eventi Keyspace, pubblicati con un prefisso di __keyspace@ <db> __ • E— Eventi chiave-evento, pubblicati con un prefisso di __keyevent@ <db> __ • g— Comandi generici e non specifici come DEL,SCADE,RENAME, e così via. • \$— Comandi Stringa • l—Elencare comandi • s— Impostare i comandi • H— Comandi hash • z— Comandi impostati ordinati • x— Eventi scaduti (eventi generati ogni volta che una chiave scade) •

Nome	Informazioni	Descrizione
		<p>e— Eventi sfrattati (eventi generati quando una chiave viene sfrattata per maxmemory)</p> <ul style="list-style-type: none">• A— Un alias per g\$shzxe <p>Può verificarsi una combinazione di questi tipi di eventi. Ad esempio, AKE indica che Redis può pubblicare notifiche di tutti i tipi di eventi.</p> <p>Non utilizzare caratteri diversi da quelli elencati sopra, altrimenti verranno restituiti messaggi di errore.</p> <p>Per impostazione predefinita, questo parametro è impostato su una stringa vuota, a indicare che la notifica per l'evento keypace è disabilitato.</p>


Nome	Informazioni	Descrizione
repl-backlog-size	<p>Impostazione predefinita: 1048576</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>La dimensione in byte del buffer di backlog del nodo principale. Il backlog viene utilizzato per la registrazione degli aggiornamenti apportati ai dati sul nodo principale e. Quando una replica di lettura si connette al nodo principale, tenta di eseguire una sincronizzazione parziale (psync), durante la quale i dati del backlog vengono applicati per il recupero tramite il nodo principale. Se psync ha esito negativo, è richiesta una sincronizzazione completa.</p> <p>Il valore minimo per questo parametro è 16384.</p> <div data-bbox="1008 1052 1507 1413"><p> Note</p><p>A partire da Redis versione 2.8.22, questo parametro si applica al cluster principale e alle repliche di lettura.</p></div>


Nome	Informazioni	Descrizione
<code>repl-backlog-ttl</code>	<p>Impostazione predefinita: 3600</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Numero di secondi durante i quali il nodo principale manterrà il buffer di backlog. A partire dalla disconnessione dell'ultimo nodo di replica, i dati nel backlog rimangono intatti fino alla scadenza di <code>repl-backlog-ttl</code>. Se la replica non viene connessa al nodo principale e entro questo periodo di tempo, il nodo principale rilascerà il buffer di backlog. Qualora la replica dovesse riconnettersi, dovrà eseguire una sincronizzazione completa con il nodo principale.</p> <p>Se questo parametro è impostato su 0, il buffer di backlog non verrà mai rilasciato.</p>
<code>repl-timeout</code>	<p>Impostazione predefinita: 60</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Rappresenta il periodo di timeout in secondi per:</p> <ul style="list-style-type: none">• Trasferimento di dati in blocco durante la sincronizzazione, dalla prospettiva della replica di lettura• Timeout del nodo principale dalla prospettiva della replica• Timeout della replica dalla prospettiva del nodo principale

Parametri di Redis 2.6.13

Famiglia del gruppo di parametri: redis2.6

Redis 2.6.13 è la prima versione di Redis supportata da ElastiCache. La tabella seguente mostra i parametri di Redis 2.6.13 supportati in ElastiCache.

Nome	Informazioni	Descrizione
activeresharding	<p>Impostazione predefinita: yes (sì)</p> <p>Tipo: stringa (yes/no)</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: infase di creazione</p>	<p>Determina se abilitare la caratteristica di attivazione di rehashing di Redis. 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>
appendonly	<p>Impostazione predefinita: no</p> <p>Tipo: stringa</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Abilita o disabilita la caratteristica AOF (append-only file), la quale acquisisce comandi Redis che modificano i dati nella cache e viene utilizzata per il ripristino da determinati errori relativi ai nodi.</p> <p>Il valore di default è no, a indicare che la caratteristica AOF è disattivata. Imposta questo parametro su yes (sì) per abilitare AOF.</p> <p>Per ulteriori informazioni, consulta Limitazione dell'impatto degli errori.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>La caratteristica AOF (Append-only file) non è supportata per i nodi</p> </div>

Nome	Informazioni	Descrizione
		<p>cache.t1.micro e cache.t2.*. Per nodi di questo tipo, il valore del parametro <code>appendonly</code> viene ignorato.</p> <p> Note</p> <p>Per i gruppi di replica Multi-AZ, la caratteristica AOF non è consentita.</p>
<code>appendfsync</code>	<p>Valore di default: <code>everysec</code></p> <p>Tipo: stringa</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Quando <code>appendonly</code> è impostato su <code>sì</code>, controlla la frequenza con cui il buffer di output AOF viene scritto su disco.</p> <ul style="list-style-type: none"> • <code>no</code>— Il buffer viene scaricato su disco in base alle esigenze. • <code>ognisec</code>— il buffer viene svuotato una volta al secondo. Questa è l'impostazione predefinita. • <code>Sempre</code>: il buffer viene svuotato ogni volta che i dati nel cluster vengono modificati. • <code>Appendfsync</code> non è supportato per le versioni 2.8.22 e successive.
<code>client-output-buffer-limit-normal-hard-limit</code>	<p>Impostazione Predefinita: 0</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</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>

Nome	Informazioni	Descrizione
<code>client-output-buffer-limit-normal-soft-limit</code>	Impostazione Predefinita: 0 Tipo: Integer Modificabile: sì Le modifiche diventano effettive: immediatamente	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).
<code>client-output-buffer-limit-normal-soft-seconds</code>	Impostazione Predefinita: 0 Tipo: Integer Modificabile: sì Le modifiche diventano effettive: immediatamente	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).
<code>client-output-buffer-limit-pubsub-hard-limit</code>	Impostazione predefinita: 33554432 Tipo: Integer Modificabile: sì Le modifiche diventano effettive: immediatamente	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>	Impostazione predefinita: 8388608 Tipo: Integer Modificabile: sì Le modifiche diventano effettive: immediatamente	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, ma solo se questa condizione persiste per <code>client-output-buffer-limit-pubsub-soft-seconds</code> .

Nome	Informazioni	Descrizione
<code>client-output-buffer-limit-pubsub-soft-seconds</code>	<p>Impostazione predefinita: 60</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	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.
<code>client-output-buffer-limit-slave-hard-limit</code>	<p>Valore di default: per i valori, vedere Parametri specifici del tipo di nodo di Redis</p> <p>Tipo: Integer</p> <p>Modificabile: no</p>	Per le repliche di lettura Redis: se il buffer di output di un client raggiunge il numero di byte specificato, il client verrà disconnesso.
<code>client-output-buffer-limit-slave-soft-limit</code>	<p>Valore di default: per i valori, vedere Parametri specifici del tipo di nodo di Redis</p> <p>Tipo: Integer</p> <p>Modificabile: no</p>	Per le repliche di lettura Redis: 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-slave-soft-seconds</code> .
<code>client-output-buffer-limit-slave-soft-seconds</code>	<p>Impostazione predefinita: 60</p> <p>Tipo: Integer</p> <p>Modificabile: no</p>	Per le repliche di lettura Redis: se il buffer di output di un client rimane di <code>client-output-buffer-limit-slave-soft-limit</code> byte per un tempo maggiore del numero di secondi specificato, il client verrà disconnesso.


Nome	Informazioni	Descrizione
databases	<p>Impostazione predefinita: 16</p> <p>Tipo: Integer</p> <p>Modificabile: no</p> <p>Le modifiche diventano effettive: infase di creazione</p>	<p>Il numero di partizioni logiche in cui sono suddivisi i database. È consigliabile mantenere basso questo valore.</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>
hash-max-ziplist-entries	<p>Impostazione predefinita: 512</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>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.</p>
hash-max-ziplist-value	<p>Impostazione predefinita: 64</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>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.</p>
list-max-ziplist-entries	<p>Impostazione predefinita: 512</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Determina la quantità di memoria utilizzata per gli elenchi. Gli elenchi con un numero di voci inferiore a quello specificato vengono archiviati con una codifica speciale che consente di risparmiare spazio.</p>

Nome	Informazioni	Descrizione
<code>list-max-ziplist-value</code>	<p>Impostazione predefinita: 64</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Determina la quantità di memoria utilizzata per gli elenchi. Gli elenchi con voci di dimensioni inferiori al numero di byte specificato vengono archiviati con una codifica speciale che consente di risparmiare spazio.</p>
<code>lua-time-limit</code>	<p>Impostazione predefinita: 5000</p> <p>Tipo: Integer</p> <p>Modificabile: no</p>	<p>Il tempo di esecuzione massimo per uno script Lua, in millisecondi, prima che ElastiCache agisca in modo da interrompere lo script.</p> <p>Se <code>lua-time-limit</code> viene superato, tutti i comandi Redis restituiscono un errore nel formato <code>____-BUSY</code>. Poiché questo stato può causare interferenza con molte operazioni Redis essenziali, ElastiCache emetterà prima un comando <code>SCRIPT KILL</code>. In caso di esito negativo, ElastiCache forzerà il riavvio di Redis.</p>
<code>maxclients</code> Questo valore si applica a tutti i tipi di istanza, eccetto quelli specificati	<p>Impostazione predefinita: 65000</p> <p>Tipo: Integer</p> <p>Modificabile: no</p> <p>t2.medium di default: 20000</p> <p>Tipo: Integer</p> <p>Modificabile: no</p> <p>t2.small di default: 20000</p> <p>Tipo: Integer</p> <p>Modificabile: no</p>	<p>Il numero massimo di client che possono essere connessi alla volta.</p>

Nome	Informazioni	Descrizione
	<p>t2.micro di default: 20000</p> <p>Tipo: Integer</p> <p>Modificabile: no</p>	
	<p>t4g.micro di default: 20000</p> <p>Tipo: Integer</p> <p>Modificabile: no</p>	
	<p>t3.medium Predefinito: 65000</p> <p>Tipo: Integer</p> <p>Modificabile: no</p>	
	<p>t3.small Predefinito: 65000</p> <p>Tipo: Integer</p> <p>Modificabile: no</p>	
	<p>t3.micro di default: 20000</p> <p>Tipo: Integer</p> <p>Modificabile: no</p>	
maxmemory-policy	<p>Impostazione predefinita: volatile-lru</p> <p>Tipo: stringa</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>La policy di espulsione per le chiavi quando viene raggiunto l'utilizzo di memoria massimo.</p> <p>I valori validi sono: volatile-lru allkeys-lru volatile-random allkeys-random volatile-ttl noeviction</p> <p>Per ulteriori informazioni, consulta Using Redis as an LRU cache.</p>

Nome	Informazioni	Descrizione
<code>maxmemory-samples</code>	<p>Impostazione predefinita: 3</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Per i calcoli relativi agli elementi utilizzati meno di recente (LRU) e time-to-live (TTL), questo parametro rappresenta la dimensione e di esempio delle chiavi da controllare. Per impostazione predefinita, Redis sceglie 3 chiavi e utilizza quella usata meno di recente.</p>
<code>reserved-memory</code>	<p>Impostazione Predefinita: 0</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>La memoria totale, in byte, prenotata per l'utilizzo di elementi diversi dai dati. Per impostazione predefinita, le dimensioni del nodo Redis aumenteranno durante l'utilizzo del valore <code>maxmemory</code> del nodo. Consulta Parametri specifici del tipo di nodo di Redis. In tal caso, le prestazioni del nodo subiranno una riduzione a causa di un eccessivo scambio di pagine di memoria. Tramite l'allocazione della memoria, puoi riservare parte della memoria disponibile per attività non Redis, per consentire la riduzione della quantità di scambio di pagine.</p> <p>Questo parametro è specifico di ElastiCache e non fa parte della distribuzione Redis standard.</p> <p>Per ulteriori informazioni, consulta <code>reserved-memory-percent</code> e Gestione della memoria prenotata.</p>
<code>set-max-intset-entries</code>	<p>Impostazione predefinita: 512</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>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.</p>

Nome	Informazioni	Descrizione
<code>slave-all- ow-chaini ng</code>	Impostazione predefinita: no Tipo: stringa Modificabile: no	Determina se una replica di lettura in Redis può disporre di repliche di lettura specifiche.
<code>slowlog-l og-slower -than</code>	Impostazione predefinita: 10000 Tipo: Integer Modificabile: sì Le modifiche diventano effettive: immediatamente	Il tempo di esecuzione massimo, in microsecondi, per i comandi da registrare tramite la caratteristica Slow Log (Log lenti) di Redis.
<code>slowlog-m ax-len</code>	Impostazione predefinita: 128 Tipo: Integer Modificabile: sì Le modifiche diventano effettive: immediatamente	La lunghezza massima di Slow Log (Log lento) di Redis.
<code>tcp-keepa live</code>	Impostazione Predefinita: 0 Tipo: Integer Modificabile: sì Le modifiche diventano effettive: immediatamente	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.

 Important

Alcuni aspetti di questo parametro sono stati modificati in Redis versione 3.2.4. Per informazioni, consultare [Parametri modificati in Redis 3.2.4 \(avanzato\)](#).

Nome	Informazioni	Descrizione
timeout	<p>Impostazione Predefinita: 0</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>Il numero di secondi prima del timeout di un nodo. I valori sono:</p> <ul style="list-style-type: none"> • 0 - Non viene mai eseguita la disconnessione di un client inattivo. • 1-19 - Valori non validi. • ≥ 20 - Il numero di secondi di attesa da parte di un nodo prima della disconnessione di un client inattivo.
zset-max-ziplist-entries	<p>Impostazione predefinita: 128</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>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.</p>
zset-max-ziplist-value	<p>Impostazione predefinita: 64</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>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.</p>

Note

Se non specifichi un gruppo di parametri per il cluster Redis 2.6.13, verrà utilizzato un gruppo di parametri di default (`default.redis2.6`) per la versione del motore. Non puoi modificare i valori dei parametri nel gruppo di parametri di default, ma puoi sempre creare un gruppo di parametri personalizzato e assegnarlo al tuo cluster in qualsiasi momento.

Parametri specifici del tipo di nodo di Redis

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 i valori predefiniti dei parametri `maxmemory`, `client-output-buffer-limit-slave-hard-limit` e `client-output-buffer-limit-slave-soft-limit` per ciascun tipo di nodo. Il valore di `maxmemory` è il numero massimo di byte disponibili sul nodo per utilizzo, dati e altro. Per ulteriori informazioni consulta [Memoria disponibile](#).

Note

Il parametro `maxmemory` non può essere modificato.

Tipo di nodo	Maxmemory	Client-output-buffer-limit-slave-hard-limit	Client-output-buffer-limit-slave-soft-limit
cache.t1.micro	142606336	14260633	14260633
cache.t2.micro	581959680	58195968	58195968
cache.t2.small	1665138688	166513868	166513868
cache.t2.medium	3461349376	346134937	346134937
cache.t3.micro	536870912	53687091	53687091
cache.t3.small	1471026299	147102629	147102629
cache.t3.medium	3317862236	331786223	331786223
cache.t4g.micro	536870912	53687091	53687091
cache.t4g.small	1471026299	147102629	147102629
cache.t4g.medium	3317862236	331786223	331786223
cache.m1.small	943718400	94371840	94371840

Tipo di nodo	Maxmemory	Client-output-buffer-limit-slave-hard-limit	Client-output-buffer-limit-slave-soft-limit
cache.m1.medium	3093299200	309329920	309329920
cache.m1.large	7025459200	702545920	702545920
cache.m1.xlarge	14889779200	1488977920	1488977920
cache.m2.xlarge	17091788800	1709178880	1709178880
cache.m2.2xlarge	35022438400	3502243840	3502243840
cache.m2.4xlarge	70883737600	7088373760	7088373760
cache.m3.medium	2988441600	309329920	309329920
cache.m3.large	6501171200	650117120	650117120
cache.m3.xlarge	14260633600	1426063360	1426063360
cache.m3.2xlarge	29989273600	2998927360	2998927360
cache.m4.large	6892593152	689259315	689259315
cache.m4.xlarge	15328501760	1532850176	1532850176
cache.m4.2xlarge	31889126359	3188912636	3188912636
cache.m4.4xlarge	65257290629	6525729063	6525729063
cache.m4.10xlarge	166047614239	16604761424	16604761424
cache.m5.large	6854542746	685454275	685454275
cache.m5.xlarge	13891921715	1389192172	1389192172
cache.m5.2xlarge	27966669210	2796666921	2796666921
cache.m5.4xlarge	56116178125	5611617812	5611617812

Tipo di nodo	Maxmemory	Client-output-buffer-limit-slave-hard-limit	Client-output-buffer-limit-slave-soft-limit
cache.m5.12xlarge	168715971994	16871597199	16871597199
cache.m5.24xlarge	337500562842	33750056284	33750056284
cache.m6g.large	6854542746	685454275	685454275
cache.m6g.xlarge	13891921715	1389192172	1389192172
cache.m6g.2xlarge	27966669210	2796666921	2796666921
cache.m6g.4xlarge	56116178125	5611617812	5611617812
cache.m6g.8xlarge	111325552312	11132555231	11132555231
cache.m6g.12xlarge	168715971994	16871597199	16871597199
cache.m6g.16xlarge	225000375228	22500037523	22500037523
cache.c1.xlarge	6501171200	650117120	650117120
cache.r3.large	14470348800	1468006400	1468006400
cache.r3.xlarge	30513561600	3040870400	3040870400
cache.r3,2xlarge	62495129600	6081740800	6081740800
cache.r3,4xlarge	126458265600	12268339200	12268339200
cache.r3,8xlarge	254384537600	24536678400	24536678400
cache.r4.large	13201781556	1320178155	1320178155
cache.r4.xlarge	26898228839	2689822883	2689822883
cache.r4.2xlarge	54197537997	5419753799	5419753799
cache.r4.4xlarge	108858546586	10885854658	10885854658

Tipo di nodo	Maxmemory	Client-output-buffer-limit-slave-hard-limit	Client-output-buffer-limit-slave-soft-limit
cache.r4.8xlarge	218255432090	21825543209	21825543209
cache.r4.16xlarge	437021573120	43702157312	43702157312
cache.r5.large	14037181030	1403718103	1403718103
cache.r5.xlarge	28261849702	2826184970	2826184970
cache.r5.2xlarge	56711183565	5671118356	5671118356
cache.r5.4xlarge	113609865216	11360986522	11360986522
cache.r5.12xlarge	341206346547	34120634655	34120634655
cache.r5.24xlarge	682485973811	68248597381	68248597381
cache.r6g.large	14037181030	1403718103	1403718103
cache.r6g.xlarge	28261849702	2826184970	2826184970
cache.r6g.2xlarge	56711183565	5671118356	5671118356
cache.r6g.4xlarge	113609865216	11360986522	11360986522
cache.r6g.8xlarge	225000375228	22500037523	22500037523
cache.r6g.12xlarge	341206346547	34120634655	34120634655
cache.r6g.16xlarge	450000750456	45000075046	45000075046
cache.r6gd.xlarge	28261849702	2826184970	2826184970
cache.r6gd.2xlarge	56711183565	5671118356	5671118356
cache.r6gd.4xlarge	113609865216	11360986522	11360986522
cache.r6gd.8xlarge	225000375228	22500037523	22500037523

Tipo di nodo	Maxmemory	Client-output-buffer-limit-slave-hard-limit	Client-output-buffer-limit-slave-soft-limit
cache.r6gd.12xlarge	341206346547	34120634655	34120634655
cache.r6gd.16xlarge	450000750456	45000075046	45000075046
cache.r7g.large	14037181030	1403718103	1403718103
cache.r7g.xlarge	28261849702	2826184970	2826184970
cache.r7g.2xlarge	56711183565	5671118356	5671118356
cache.r7g.4xlarge	113609865216	11360986522	11360986522
cache.r7g.8xlarge	225000375228	22500037523	22500037523
cache.r7g.12xlarge	341206346547	34120634655	34120634655
cache.r7g.16xlarge	450000750456	45000075046	45000075046
cache.m7g.large	6854542746	685454275	685454275
cache.m7g.xlarge	13891921715	1389192172	1389192172
cache.m7g.2xlarge	27966669210	2796666921	2796666921
cache.m7g.4xlarge	56116178125	5611617812	5611617812
cache.m7g.8xlarge	111325552312	11132555231	11132555231
cache.m7g.12xlarge	168715971994	16871597199	16871597199
cache.m7g.16xlarge	225000375228	22500037523	22500037523
cache.c7gn.large	3317862236	1403718103	1403718103
cache.c7gn.xlarge	6854542746	2826184970	2826184970
cache.c7gn.2xlarge	13891921715	5671118356	5671118356

Tipo di nodo	Maxmemory	Client-output-buffer-limit-slave-hard-limit	Client-output-buffer-limit-slave-soft-limit
cache.c7gn.4xlarge	27966669210	11360986522	11360986522
cache.c7gn.8xlarge	56116178125	22500037523	22500037523
cache.c7gn.12xlarge	84357985997	34120634655	34120634655
cache.c7gn.16xlarge	113609865216	45000075046	45000075046

Note

Tutti i tipi di istanza della generazione corrente sono creati in Amazon Virtual Private Cloud VPC per impostazione predefinita.

Le istanze T1 non supportano la funzione Multi-AZ.

Le istanze T1 e T2 non supportano la caratteristica AOF (append-only file) Redis.

Le variabili di configurazione Redis `appendonly` e `appendfsync` non sono supportate su Redis versione 2.8.22 e successive.

Scalabilità per Redis ElastiCache

Scalabilità ElastiCache senza server

ElastiCache Serverless si adatta automaticamente al traffico del carico di lavoro man mano che aumenta o diminuisce. Per ogni cache ElastiCache Serverless, monitora ElastiCache continuamente l'utilizzo di risorse come CPU, memoria e rete. Quando una di queste risorse è limitata, ElastiCache Serverless esegue la scalabilità orizzontale aggiungendo un nuovo shard e ridistribuendo i dati sul nuovo shard, senza tempi di inattività dell'applicazione. È possibile monitorare le risorse utilizzate dalla cache monitorando la `BytesUsedForCache` metrica per l'archiviazione dei dati nella `CloudWatch cache` e (`ECPU`) per l'utilizzo dell'elaborazione. `ElastiCacheProcessingUnits`

Impostazione dei limiti di dimensionamento per gestire i costi

Puoi scegliere di configurare un utilizzo massimo dell'archiviazione di dati e delle ECPU al secondo per controllare i costi della cache. In questo modo ti assicuri che l'utilizzo della cache non superi mai il limite massimo configurato.

Se imposti un dimensionamento massimo, l'applicazione potrebbe subire una riduzione delle prestazioni quando la cache raggiunge il limite massimo. Quando imposti un valore massimo di archiviazione dei dati nella cache e lo spazio di archiviazione dei dati della cache raggiunge il massimo, ElastiCache inizierà a rimuovere i dati dalla cache con un set Time-To-Live (TTL), utilizzando la logica LRU. Se non sono disponibili dati che possono essere sottoposti a espulsione, le richieste di scrittura di dati aggiuntivi restituiscono un messaggio di errore di memoria insufficiente (OOM). Quando imposti un valore massimo di ECPU/secondo e l'utilizzo di calcolo del carico di lavoro supera questo valore, inizieranno a limitare le richieste Redis. ElastiCache

Se imposti un limite massimo su `BytesUsedForCache` o `ElastiCacheProcessingUnits`, ti consigliamo vivamente di impostare un CloudWatch allarme a un valore inferiore al limite massimo in modo da ricevere una notifica quando la cache funziona in prossimità di questi limiti. Ti consigliamo di impostare l'allarme al 75% del limite massimo configurato. Consulta la documentazione su come configurare gli CloudWatch allarmi.

Prescalabilità con Serverless ElastiCache

ElastiCache Prescalabilità senza server

Con il predimensionamento, chiamato anche preriscaldamento, è possibile impostare limiti minimi supportati per la cache. ElastiCache È possibile impostare questi valori minimi per le unità di ElastiCache elaborazione (ECPU) al secondo o per l'archiviazione dei dati. Ciò può essere utile in preparazione agli eventi di scalabilità previsti. Ad esempio, se una società di giochi prevede un aumento di 5 volte degli accessi entro il primo minuto dal lancio del nuovo gioco, può predisporre la cache per questo significativo picco di utilizzo.

È possibile eseguire la prescalabilità utilizzando la ElastiCache console, la CLI o l'API. ElastiCache Serverless aggiorna le ECPU disponibili al secondo nella cache entro 60 minuti e invia una notifica di evento quando viene completato l'aggiornamento del limite minimo.

Come funziona la prescalabilità

Quando il limite minimo per ECPU/secondo o l'archiviazione dei dati viene aggiornato tramite console, CLI o API, il nuovo limite è disponibile entro 1 ora. ElastiCache Serverless supporta 30.000

ECPUS/secondo su una cache vuota e fino a 90.000 ECPUS/sec quando si utilizza la funzione Read from Replica. ElastiCache può raddoppiare le ECPU al secondo ogni 10-12 minuti. Questa velocità di scalabilità è sufficiente per la maggior parte dei carichi di lavoro. Se prevedi che un evento di scalabilità imminente potrebbe superare questa frequenza, ti consigliamo di impostare le ECPU/secondo minime sul picco di ECPUS/sec previsto almeno 60 minuti prima dell'evento di picco. In caso contrario, l'applicazione potrebbe riscontrare una latenza e una limitazione delle richieste elevate.

Una volta completato l'aggiornamento del limite minimo, ElastiCache Serverless inizierà a contabilizzare il nuovo numero minimo di ECPU al secondo o il nuovo spazio di archiviazione minimo. Ciò si verifica anche se l'applicazione non esegue richieste sulla cache o se l'utilizzo dell'archiviazione dei dati è inferiore al minimo. Quando si abbassa il limite minimo rispetto all'impostazione corrente, l'aggiornamento è immediato, quindi ElastiCache Serverless inizierà immediatamente a misurare il nuovo limite minimo.

Note

- Quando imposti un limite di utilizzo minimo, ti viene addebitato il costo di tale limite anche se l'utilizzo effettivo è inferiore al limite di utilizzo minimo. All'utilizzo dell'ECPU o dell'archiviazione dei dati che supera il limite di utilizzo minimo viene addebitata la tariffa normale. Ad esempio, se imposti un limite di utilizzo minimo di 100.000 ECPU al secondo, ti verranno addebitati almeno 1,224 USD all'ora (utilizzando i prezzi delle ECPU in us-east-1), anche se l'utilizzo è inferiore al minimo stabilito.
- ElastiCache Serverless supporta la scala minima richiesta a livello aggregato sulla cache. ElastiCache Serverless supporta anche un massimo di 30.000 ECPU/secondo per slot (90.000 ECPU/secondo quando si utilizza Read from Replica utilizzando connessioni READONLY). Come best practice, l'applicazione dovrebbe garantire che la distribuzione delle chiavi tra gli slot Redis e il traffico tra le chiavi siano il più uniformi possibile.

Impostazione dei limiti di scalabilità utilizzando la console e AWS CLI

Impostazione dei limiti di scalabilità tramite la console AWS

1. Accedi AWS Management Console e apri la ElastiCache console all'[indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Nel riquadro di navigazione scegli il motore in esecuzione sulla cache da modificare.
3. Viene visualizzato un elenco di cache che eseguono il motore scelto.

4. Scegli la cache da modificare facendo clic sul pulsante di opzione a sinistra del nome della cache.
5. Scegliere Actions (Operazioni), quindi Modify (Modifica).
6. In Limiti di utilizzo, imposta i limiti di memoria o elaborazione appropriati.
7. Fai clic su Visualizza l'anteprima delle modifiche e quindi su Salva le modifiche.

Impostazione dei limiti di scalabilità utilizzando il AWS CLI

Per modificare i limiti di scalabilità utilizzando la CLI, utilizza l' `modify-serverless-cache` API.

Linux:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> \  
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},  
ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

Windows:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> ^  
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},  
ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

Rimozione dei limiti di dimensionamento tramite la CLI

Per rimuovere i limiti di scalabilità utilizzando la CLI, imposta il parametro Limite minimo e massimo su 0.

Linux:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> \  
--cache-usage-limits 'DataStorage={Minimum=0,Maximum=0,Unit=GB},  
ECPUPerSecond={Minimum=0,Maximum=0}'
```

Windows:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> ^  
--cache-usage-limits 'DataStorage={Minimum=0,Maximum=0,Unit=GB},  
ECPUPerSecond={Minimum=0,Maximum=0}'
```

Scalabilità per cluster progettati autonomamente da Redis ElastiCache

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 da solo la tua cache, è necessario disporre dell'hardware sufficiente per soddisfare i picchi della domanda e questo può comportare costi cospicui. Con Amazon ElastiCache puoi scalare per soddisfare la domanda attuale, pagando solo per ciò che usi. ElastiCache ti consente di scalare la cache in base alla domanda.

I contenuti seguenti ti danno la possibilità di trovare l'argomento adatto alle operazioni di scalabilità di tuo interesse.

Dimensionamento di cluster Redis

Azione	Redis (modalità cluster disabilitata)	Redis (modalità cluster abilitata)
Dimensionamento verticale	Rimozione di nodi da un cluster	Ridimensionamento dei cluster in Redis (modalità cluster attivata)
Dimensionamento orizzontale	Aggiunta di nodi a un cluster	Reshard online e ribilanciamento della partizione per Redis (modalità cluster abilitata)
Modifica dei tipi di nodo	<p>Verso un tipo di nodo più grande:</p> <ul style="list-style-type: none"> Dimensionamento verso l'alto dei cluster a nodo singolo per Redis (modalità cluster disabilitata) Dimensionamento verso l'alto di cluster Redis con repliche 	Ridimensionamento verticale online tramite la modifica del tipo di nodo

Azione	Redis (modalità cluster disabilitata)	Redis (modalità cluster abilitata)
	<p>Verso un tipo di nodo più piccolo:</p> <ul style="list-style-type: none">• Dimensionamento verso il basso di cluster Redis a nodo singolo• Dimensionamento verso il basso delle risorse dei cluster Redis con repliche	
Modifica del numero di gruppi del nodo	Non supportato per cluster Redis (modalità cluster disabilitata)	Ridimensionamento dei cluster in Redis (modalità cluster attivata)

Argomenti

- [Ridimensionamento dei cluster per Redis \(modalità cluster disabilitata\)](#)
- [Ridimensionamento dei cluster in Redis \(modalità cluster attivata\)](#)

Ridimensionamento dei cluster per Redis (modalità cluster disabilitata)

I cluster Redis (modalità cluster disabilitata) possono essere un cluster a nodo singolo con 0 shard o cluster a più nodi con 1 shard. I cluster a nodo singolo usano un nodo per entrambe le operazioni di lettura e scrittura. I cluster a più nodi hanno sempre 1 nodo come nodo primario lettura/scrittura con da 0 a 5 nodi di replica in sola lettura.

Indice

- [Ridimensionamento dei cluster a nodo singolo per Redis \(modalità cluster disabilitata\)](#)
 - [Dimensionamento verso l'alto dei cluster a nodo singolo per Redis \(modalità cluster disabilitata\)](#)
 - [Dimensionamento verso l'alto dei cluster a nodo singolo per Redis \(Modalità cluster disabilitata\) \(Console\)](#)
 - [Dimensionamento verso l'alto di cluster di cache Redis a nodo singolo \(AWS CLI\)](#)
 - [Dimensionamento verso l'alto di cluster di cache Redis a nodo singolo \(API ElastiCache\)](#)
 - [Dimensionamento verso il basso di cluster Redis a nodo singolo](#)
 - [Riduzione verticale di cluster Redis a nodo singolo \(console\)](#)
 - [Dimensionamento verso il basso di cluster di cache Redis a nodo singolo \(AWS CLI\)](#)
 - [Riduzione verticale di cluster di cache Redis a nodo singolo \(API ElastiCache\)](#)
- [Ridimensionamento dei cluster Redis \(modalità cluster disabilitata\) con nodi di replica](#)
 - [Dimensionamento verso l'alto di cluster Redis con repliche](#)
 - [Dimensionamento verso il basso delle risorse dei cluster Redis con repliche](#)
 - [Aumento della capacità di lettura](#)
 - [Diminuzione della capacità di lettura](#)

Ridimensionamento dei cluster a nodo singolo per Redis (modalità cluster disabilitata)

I nodi Redis (modalità cluster disabilitata) devono essere abbastanza grandi da contenere tutti i dati della cache oltre l'overhead Redis. Per modificare la capacità dei dati del cluster Redis (modalità cluster disabilitata), è necessario effettuare il dimensionamento verticale; dimensionare verso l'alto a un tipo di nodo più grande per aumentare la capacità dei dati o dimensionare verso il basso a un tipo di nodo più piccolo per ridurre la capacità dei dati.

Il processo di dimensionamento verso l'alto di ElastiCache for Redis è progettato per cercare di mantenere i dati esistenti e, a tale scopo, richiede una corretta replica Redis. Per i cluster Redis (modalità cluster disabilitata), consigliamo di rendere disponibile memoria libera sufficiente per Redis.

Non è possibile effettuare una partizione dei dati su più cluster Redis (modalità cluster disabilitata). Tuttavia, se è necessario solo aumentare o diminuire la capacità di lettura del cluster, è possibile creare un cluster Redis (modalità cluster disabilitata) con i nodi di replica e aggiungere o rimuovere repliche. Per creare un cluster Redis (modalità cluster disabilitata) con i nodi di replica utilizzando i cluster di cache Redis a nodo singolo come cluster primario, consultare [Creazione di un cluster Redis \(modalità cluster disabilitata\) \(Console\)](#).

Dopo aver creato il cluster con le repliche, è possibile aumentare la capacità di lettura aggiungendo repliche di lettura. In seguito, se necessario, è possibile ridurre la capacità di lettura rimuovendo le repliche di lettura. Per ulteriori informazioni, consultare [Aumento della capacità di lettura](#) o [Diminuzione della capacità di lettura](#).

Oltre a dare la possibilità di dimensionare la capacità di lettura, i cluster Redis (modalità cluster disabilitata) con repliche offrono altri vantaggi per il business. Per ulteriori informazioni, consulta [Alta disponibilità utilizzando gruppi di replica](#).

Important

Se il gruppo di parametri utilizza `reserved-memory` per riservare memoria per un overhead Redis, prima di iniziare il dimensionamento verificare di disporre di un gruppo di parametri personalizzato che riserva una quantità corretta di memoria per il nuovo tipo di nodo. In alternativa, è possibile modificare un gruppo di parametri personalizzato in modo che utilizzi `reserved-memory-percent` e il gruppo di parametri per il nuovo cluster.

Se stai utilizzando `reserved-memory-percent`, questa operazione non è necessaria.

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

Argomenti

- [Dimensionamento verso l'alto dei cluster a nodo singolo per Redis \(modalità cluster disabilitata\)](#)
- [Dimensionamento verso il basso di cluster Redis a nodo singolo](#)

Dimensionamento verso l'alto dei cluster a nodo singolo per Redis (modalità cluster disabilitata)

Quando si effettua il dimensionamento verso l'alto di un cluster Redis a nodo singolo, ElastiCache esegue il processo seguente, indipendentemente dal fatto che si utilizzi la console ElastiCache, l'AWS CLI o l'API ElastiCache.

1. Un nuovo cluster di cache con un nuovo tipo di nodo viene avviato nella stessa zona di disponibilità del cluster di cache esistente.
2. I dati di cache presenti nel cluster di cache esistente vengono copiati sul nuovo cluster di cache. La durata di questo processo dipende dal tipo di nodo e dalla quantità di dati presenti nel cluster di cache.
3. Le attività di lettura e scrittura vengono riprese utilizzando il nuovo cluster di cache. Poiché i nuovi endpoint del cluster di cache sono gli stessi del cluster di cache precedente, non è necessario aggiornare gli endpoint nell'applicazione. Noterai una breve interruzione (qualche secondo) di letture e scritture dal nodo primario mentre la voce DNS viene aggiornata.
4. ElastiCache elimina il cluster di cache precedente. Noterai una breve interruzione (pochi secondi) di letture e scritture dal vecchio nodo perché le connessioni al vecchio nodo verranno disconnesse.

Note

Per i cluster che eseguono il tipo di nodo r6gd, è possibile dimensionare solo le dimensioni dei nodi all'interno della famiglia di nodi r6gd.

Come illustrato nella tabella seguente, l'operazione di dimensionamento verso l'alto di Redis si blocca in caso vi sia un upgrade del motore pianificato per la finestra di manutenzione successiva. Per ulteriori informazioni sulla manutenzione Windows, consultare [Gestione della manutenzione](#).

Operazioni Redis bloccate

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

Operazioni in sospeso	Operazioni bloccate
	Aggiornamento immediato del motore

Se hai un'operazione in sospeso che ti blocca, puoi svolgere una delle seguenti operazioni.

- Programmare l'operazione di dimensionamento verso l'alto del motore Redis per una finestra di manutenzione successiva, deselegando la casella di controllo Apply immediately (Applica immediatamente) (con la CLI utilizzare `--no-apply-immediately`; con l'API, ci si può avvalere di `ApplyImmediately=false`).
- Attendere una finestra di manutenzione successiva per l'operazione di dimensionamento verso l'alto di Redis.
- Aggiungere l'aggiornamento del motore Redis a questa modifica del cluster di cache con la casella di controllo Apply Immediately (Applica immediatamente) selezionata (on la CLI utilizzare `--apply-immediately`; con l'API, ci si può avvalere di `ApplyImmediately=true`). In questo modo si sblocca l'operazione di dimensionamento verso l'alto per eseguire immediatamente l'aggiornamento automatico del motore.

Puoi aumentare un cluster Redis a nodo singolo (modalità cluster disabilitata) utilizzando la console ElastiCache aCche,AWS CLlo API ElastiCache.

Important

Se il gruppo di parametri utilizza `reserved-memory` per riservare memoria per un overhead Redis, prima di iniziare il dimensionamento verificare di disporre di un gruppo di parametri personalizzato che riserva una quantità corretta di memoria per il nuovo tipo di nodo. In alternativa, è possibile modificare un gruppo di parametri personalizzato in modo che utilizzi `reserved-memory-percent` e il gruppo di parametri per il nuovo cluster.

Se stai utilizzando `reserved-memory-percent`, questa operazione non è necessaria.

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

Dimensionamento verso l'alto dei cluster a nodo singolo per Redis (Modalità cluster disabilitata) (Console)

La procedura seguente descrive come eseguire un dimensionamento verso l'alto di un cluster Redis a nodo singolo utilizzando la console di gestione ElastiCache. Durante questo processo, il cluster Redis continuerà ad elaborare le richieste con tempi di inattività minimi.

Per il dimensionamento verso l'alto di un cluster Redis a nodo singolo (console)

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dal riquadro di navigazione scegli Cluster Redis.
3. Dall'elenco dei cluster, scegliere il cluster da dimensionare verso l'alto (deve eseguire il motore Redis, non il motore Redis in cluster).
4. Scegli Modifica.
5. Nella procedura guidata Modify Cluster (Modifica cluster):
 - a. Scegliere il tipo di nodo a cui dimensionare dall'elenco Node type (Tipo di nodo).
 - b. Se stai utilizzando `reserved-memory` per gestire la tua memoria, dall'elenco Parameter Group (Gruppo di parametri), seleziona il gruppo di parametri personalizzati che riserva la quantità corretta di memoria per il nuovo tipo di nodo.
6. Per eseguire subito il processo di dimensionamento verso l'alto, scegliere la casella Apply immediately (Applica immediatamente). Se non viene selezionata la casella Apply immediately (Applica immediatamente), il processo di dimensionamento verso l'alto viene eseguito nel corso della successiva finestra di manutenzione del cluster.
7. Scegli Modifica.

Se, nel passaggio precedente, è stato scelto Apply immediately (Applica immediatamente), lo stato del cluster diventa modifica in corso. Quando lo stato cambia in disponibile, la modifica è completa ed è possibile iniziare a utilizzare il nuovo cluster.

Dimensionamento verso l'alto di cluster di cache Redis a nodo singolo (AWS CLI)

La procedura seguente descrive come eseguire un dimensionamento verso l'alto di un cluster di cache Redis a nodo singolo utilizzando la console di gestione AWS CLI. Durante questo processo, il cluster Redis continuerà ad elaborare le richieste con tempi di inattività minimi.

Per aumentare un cluster di cache Redis a nodo singolo (AWS CLI)

1. Stabilire per quali tipi di nodi è possibile eseguire il dimensionamento verso l'alto eseguendo il comando `list-allowed-node-type-modifications` dell'AWS CLI con il seguente parametro.

- `--cache-cluster-id`

Per Linux, macOS o Unix:

```
aws elasticache list-allowed-node-type-modifications \  
  --cache-cluster-id my-cache-cluster-id
```

Per Windows:

```
aws elasticache list-allowed-node-type-modifications ^  
  --cache-cluster-id my-cache-cluster-id
```

L'output del comando in alto è simile al seguente (in formato JSON).

```
{  
  "ScaleUpModifications": [  
    "cache.m3.2xlarge",  
    "cache.m3.large",  
    "cache.m3.xlarge",  
    "cache.m4.10xlarge",  
    "cache.m4.2xlarge",  
    "cache.m4.4xlarge",  
    "cache.m4.large",  
    "cache.m4.xlarge",  
    "cache.r3.2xlarge",  
    "cache.r3.4xlarge",  
    "cache.r3.8xlarge",  
    "cache.r3.large",  
    "cache.r3.xlarge"  
  ]  
  "ScaleDownModifications": [  
    "cache.t2.micro",  
    "cache.t2.small ",  
    "cache.t2.medium ",  
    "cache.t1.small ",  
  ]  
}
```

```
    ],  
  }  
}
```

Per ulteriori informazioni, consulta [list-allowed-node-type-modifications](#) nella documentazione di riferimento AWS CLI.

2. Modifica il cluster di cache esistente specificando il cluster di cache da dimensionare verso l'alto e il nuovo tipo di nodo più grande, utilizzando il comando AWS CLI dell'API `modify-cache-cluster` e i seguenti parametri.

- `--cache-cluster-id` : Il nome del cluster di cache che stai dimensionando verso l'alto.
- `--cache-node-type` : Il nuovo tipo di nodo con il quale intendi dimensionare il cluster di cache. Questo valore deve essere uno dei tipi di nodi restituiti dal comando `list-allowed-node-type-modifications` nella fase 1.
- `--cache-parameter-group-name` – [Facoltativo] Utilizzare questo parametro se si usa `reserved-memory` per gestire la memoria prenotata del cluster. Specificare un gruppo di parametri di cache personalizzato che riserva la quantità di memoria corretta per il nuovo tipo di nodo. Se si sta utilizzando `reserved-memory-percent` è possibile omettere questo parametro.
- `--apply-immediately` : Consente di applicare immediatamente il dimensionamento verso l'alto. Per posporre il processo di dimensionamento alla finestra di manutenzione successiva del cluster, utilizzare il parametro `--no-apply-immediately`.

Per Linux, macOS o Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-redis-cache-cluster \  
  --cache-node-type cache.m3.xlarge \  
  --cache-parameter-group-name redis32-m2-x1 \  
  --apply-immediately
```

Per Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-redis-cache-cluster ^  
  --cache-node-type cache.m3.xlarge ^  
  --cache-parameter-group-name redis32-m2-x1 ^
```

```
--apply-immediately
```

L'output del comando in alto è simile al seguente (in formato JSON).

```
{
  "CacheCluster": {
    "Engine": "redis",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "CacheParameterGroupName": "default.redis6.x",
      "ParameterApplyStatus": "in-sync"
    },
    "SnapshotRetentionLimit": 1,
    "CacheClusterId": "my-redis-cache-cluster",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1,
    "SnapshotWindow": "00:00-01:00",
    "CacheClusterCreateTime": "2017-02-21T22:34:09.645Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "modifying",
    "PreferredAvailabilityZone": "us-west-2a",
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "CacheSubnetGroupName": "default",
    "EngineVersion": "6.0",
    "PendingModifiedValues": {
      "CacheNodeType": "cache.m3.2xlarge"
    },
    "PreferredMaintenanceWindow": "tue:11:30-tue:12:30",
    "CacheNodeType": "cache.m3.medium",
    "DataTiering": "disabled"
  }
}
```

Per ulteriori informazioni, consulta [modify-cache-cluster](#) nella documentazione di riferimento AWS CLI.

3. Se è stato utilizzato `--apply-immediately`, monitorare lo stato del nuovo cluster di cache utilizzando il comando AWS CLI `describe-cache-clusters` con il seguente parametro. Quando lo stato cambia in disponibile, è possibile iniziare a utilizzare il nuovo cluster di cache più grande.

- `--cache-cache cluster-id` : Il nome del cluster di cache Redis a nodo singolo. Utilizzare questo parametro per descrivere un cluster di cache particolare piuttosto che tutti i cluster di cache.

```
aws elasticache describe-cache-clusters --cache-cluster-id my-redis-cache-cluster
```

Per ulteriori informazioni, consulta [describe-cache-clusters](#) nella documentazione di riferimento AWS CLI.

Dimensionamento verso l'alto di cluster di cache Redis a nodo singolo (API ElastiCache)

La procedura seguente descrive come eseguire un dimensionamento verso l'alto di un cluster di cache Redis a nodo singolo utilizzando l'API ElastiCache. Durante questo processo, il cluster Redis continuerà ad elaborare le richieste con tempi di inattività minimi.

Per il dimensionamento verso l'alto di un cluster di cache Redis a nodo singolo (API ElastiCache)

1. Stabilire per quali tipi di nodi è possibile eseguire il dimensionamento verso l'alto eseguendo l'operazione `ListAllowedNodeTypeModifications` dell'API ElastiCache con il seguente parametro.
 - `CacheClusterId` : Il nome del cluster di cache Redis a nodo singolo a cui desideri effettuare il dimensionamento verso l'alto.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListAllowedNodeTypeModifications  
&CacheClusterId=MyRedisCacheCluster  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta [ListAllowedNodeTypeModifications](#) nella Guida di riferimento delle API di Amazon ElastiCache.

2. Modifica il cluster di cache esistente specificando il cluster di cache da aumentare e il nuovo tipo di nodo più grande, utilizzando l'operazione ElastiCache dell'API `ModifyCacheCluster` e i seguenti parametri.
 - `CacheClusterId` : Il nome del cluster di cache che stai ridimensionando verso l'alto.
 - `CacheNodeType` : Il nuovo tipo di nodo più grande a cui desideri dimensionare verso il basso il cluster di cache. Questo valore deve essere uno dei tipi di nodo restituiti dall'operazione `ListAllowedNodeTypeModifications` nella fase 1.
 - `CacheParameterGroupName` – [Facoltativo] Utilizzare questo parametro se si usa `reserved-memory` per gestire la memoria prenotata del cluster. Specificare un gruppo di parametri di cache personalizzato che riserva la quantità di memoria corretta per il nuovo tipo di nodo. Se si sta utilizzando `reserved-memory-percent` è possibile omettere questo parametro.
 - `ApplyImmediately` : Imposta su `true` per eseguire immediatamente il processo di dimensionamento verso l'alto. Per posporre il processo di dimensionamento alla finestra di manutenzione successiva del cluster, utilizzare `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyCacheCluster  
  &ApplyImmediately=true  
  &CacheClusterId=MyRedisCacheCluster  
  &CacheNodeType=cache.m3.xlarge  
  &CacheParameterGroupName redis32-m2-x1  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta [ModifyCacheCluster](#) nella Guida di riferimento delle API di Amazon ElastiCache.

3. Se è stato utilizzato `ApplyImmediately=true`, controllare lo stato del nuovo cluster di cache utilizzando l'operazione ElastiCache dell'API `DescribeCacheClusters` con il seguente parametro. Quando lo stato cambia in disponibile, è possibile iniziare a utilizzare il nuovo cluster di cache più grande.

- `CacheClusterId` : Il nome del cluster di cache Redis a nodo singolo. Utilizzare questo parametro per descrivere un cluster di cache particolare piuttosto che tutti i cluster di cache.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheClusters  
  &CacheClusterId=MyRedisCacheCluster  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta [DescribeCacheClusters](#) nella Guida di riferimento delle API di Amazon ElastiCache.

Dimensionamento verso il basso di cluster Redis a nodo singolo

Le sezioni seguenti descrivono in modo dettagliato come dimensionare verso il basso un cluster Redis a nodo singolo a un tipo di nodo più piccolo. La garanzia che il nuovo tipo di nodo più piccolo sia abbastanza grande da accogliere tutti i dati e l'overhead Redis è molto importante per il successo a lungo termine del nuovo cluster Redis. Per ulteriori informazioni, consulta [Verifica della disponibilità di memoria sufficiente per creare uno snapshot Redis](#).

Note

Per i cluster che eseguono il tipo di nodo r6gd, è possibile dimensionare solo le dimensioni dei nodi all'interno della famiglia di nodi r6gd.

Argomenti

- [Riduzione verticale di cluster Redis a nodo singolo \(console\)](#)
- [Dimensionamento verso il basso di cluster di cache Redis a nodo singolo \(AWS CLI\)](#)
- [Riduzione verticale di cluster di cache Redis a nodo singolo \(API ElastiCache\)](#)

Riduzione verticale di cluster Redis a nodo singolo (console)

La procedura seguente descrive in modo dettagliato come dimensionare verso il basso un cluster Redis a nodo singolo a un tipo di nodo più piccolo utilizzando la console ElastiCache.

Important

Se il gruppo di parametri utilizza `reserved-memory` per riservare memoria per un overhead Redis, prima di iniziare il dimensionamento verificare di disporre di un gruppo di parametri personalizzato che riserva una quantità corretta di memoria per il nuovo tipo di nodo. In alternativa, è possibile modificare un gruppo di parametri personalizzato in modo che utilizzi `reserved-memory-percent` e il gruppo di parametri per il nuovo cluster.

Se stai utilizzando `reserved-memory-percent`, questa operazione non è necessaria. Per ulteriori informazioni, consulta [Gestione della memoria prenotata](#).

Per la riduzione verticale di un cluster Redis a nodo singolo (console)

1. Verificare che il dimensionamento a un tipo di nodo più piccolo sia adeguato ai dati e all'overhead.
2. Se il gruppo di parametri utilizza `reserved-memory` per riservare memoria per un overhead Redis, è necessario avere la garanzia di disporre di un gruppo di parametri personalizzato per riservare una quantità corretta di memoria per il nuovo tipo di nodo.

In alternativa, puoi modificare un gruppo di parametri personalizzato per utilizzare `reserved-memory-percent`. Per ulteriori informazioni, consulta [Gestione della memoria prenotata](#).

3. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
4. Nell'elenco dei cluster, scegliere quello da diminuire. Questo cluster deve eseguire un motore Redis e non il motore Redis in cluster.
5. Scegli Modifica.
6. Nella procedura guidata Modify Cluster (Modifica cluster):
 - a. Scegliere il tipo di nodo che si desidera diminuire dall'elenco Node type (Tipo di nodo).
 - b. Se stai utilizzando `reserved-memory` per gestire la tua memoria, dall'elenco Parameter Group (Gruppo di parametri), seleziona il gruppo di parametri personalizzati che riserva la quantità corretta di memoria per il nuovo tipo di nodo.
7. Per eseguire subito il processo di dimensionamento, scegliere la casella Apply immediately (Applica immediatamente). Se non viene selezionata la casella Apply immediately (Applica immediatamente), il processo di dimensionamento viene eseguito nel corso della successiva finestra di manutenzione del cluster.
8. Scegli Modifica.
9. Quando lo stato del cluster cambia da modifica in corso a disponibile, il cluster è stato dimensionato a un nuovo tipo di nodo. Non è necessario aggiornare gli endpoint nell'applicazione.

Dimensionamento verso il basso di cluster di cache Redis a nodo singolo (AWS CLI)

La procedura seguente descrive come eseguire un dimensionamento verso il basso di un cluster di cache Redis a nodo singolo utilizzando la console di gestione AWS CLI.

Per ridurre un cluster di cache Redis a nodo singolo (AWS CLI)

1. Stabilire per quali tipi di nodi è possibile eseguire il ridimensionamento verso il basso eseguendo il comando AWS CLI `list-allowed-node-type-modifications` con il seguente parametro.

- `--cache-cluster-id`

Per Linux, macOS o Unix:

```
aws elasticache list-allowed-node-type-modifications \  
  --cache-cluster-id my-cache-cluster-id
```

Per Windows:

```
aws elasticache list-allowed-node-type-modifications ^  
  --cache-cluster-id my-cache-cluster-id
```

L'output del comando in alto è simile al seguente (in formato JSON).

```
{  
  "ScaleUpModifications": [  
    "cache.m3.2xlarge",  
    "cache.m3.large",  
    "cache.m3.xlarge",  
    "cache.m4.10xlarge",  
    "cache.m4.2xlarge",  
    "cache.m4.4xlarge",  
    "cache.m4.large",  
    "cache.m4.xlarge",  
    "cache.r3.2xlarge",  
    "cache.r3.4xlarge",  
    "cache.r3.8xlarge",  
    "cache.r3.large",  
    "cache.r3.xlarge"  
  ]  
  "ScaleDownModifications": [  
    "cache.t2.micro",  
    "cache.t2.small ",  
    "cache.t2.medium ",  
    "cache.t1.small ",  
  ]  
}
```

```
    ],  
  }  
}
```

Per ulteriori informazioni, consulta [list-allowed-node-type-modifications](#) nella documentazione di riferimento AWS CLI.

2. Modifica il cluster di cache esistente specificando il cluster di cache da dimensionare verso il basso e il nuovo tipo di nodo più piccolo, utilizzando il comando AWS CLI dell'API `modify-cache-cluster` e i seguenti parametri.
 - `--cache-cluster-id` : Il nome del cluster di cache che stai riducendo verticalmente
 - `--cache-node-type` : Il nuovo tipo di nodo con il quale intendi dimensionare il cluster di cache. Questo valore deve essere uno dei tipi di nodi restituiti dal comando `list-allowed-node-type-modifications` nella fase 1.
 - `--cache-parameter-group-name` – [Facoltativo] Utilizzare questo parametro se si usa `reserved-memory` per gestire la memoria prenotata del cluster. Specificare un gruppo di parametri di cache personalizzato che riserva la quantità di memoria corretta per il nuovo tipo di nodo. Se si sta utilizzando `reserved-memory-percent` è possibile omettere questo parametro.
 - `--apply-immediately` : Causa l'applicazione immediata del processo di riduzione verticale. Per postporre il processo di dimensionamento alla finestra di manutenzione successiva del cluster, utilizzare il parametro `--no-apply-immediately`.

Per Linux, macOS o Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-redis-cache-cluster \  
  --cache-node-type cache.m3.xlarge \  
  --cache-parameter-group-name redis32-m2-x1 \  
  --apply-immediately
```

Per Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-redis-cache-cluster ^  
  --cache-node-type cache.m3.xlarge ^  
  --cache-parameter-group-name redis32-m2-x1 ^
```

```
--apply-immediately
```

L'output del comando in alto è simile al seguente (in formato JSON).

```
{
  "CacheCluster": {
    "Engine": "redis",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "CacheParameterGroupName": "default.redis6.x",
      "ParameterApplyStatus": "in-sync"
    },
    "SnapshotRetentionLimit": 1,
    "CacheClusterId": "my-redis-cache-cluster",
    "CacheSecurityGroups": [],
    "NumCacheNodes": 1,
    "SnapshotWindow": "00:00-01:00",
    "CacheClusterCreateTime": "2017-02-21T22:34:09.645Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "modifying",
    "PreferredAvailabilityZone": "us-west-2a",
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "CacheSubnetGroupName": "default",
    "EngineVersion": "6.0",
    "PendingModifiedValues": {
      "CacheNodeType": "cache.m3.2xlarge"
    },
    "PreferredMaintenanceWindow": "tue:11:30-tue:12:30",
    "CacheNodeType": "cache.m3.medium",
    "DataTiering": "disabled"
  }
}
```

Per ulteriori informazioni, consulta [modify-cache-cluster](#) nella documentazione di riferimento AWS CLI.

3. Se è stato utilizzato `--apply-immediately`, monitorare lo stato del nuovo cluster di cache utilizzando il comando AWS CLI `describe-cache-clusters` con il seguente parametro. Quando lo stato cambia in disponibile, è possibile iniziare a utilizzare il nuovo cluster di cache più grande.

- `--cache-cache cluster-id` : Il nome del cluster di cache Redis a nodo singolo. Utilizzare questo parametro per descrivere un cluster di cache particolare piuttosto che tutti i cluster di cache.

```
aws elasticache describe-cache-clusters --cache-cluster-id my-redis-cache-cluster
```

Per ulteriori informazioni, consulta [describe-cache-clusters](#) nella documentazione di riferimento AWS CLI.

Riduzione verticale di cluster di cache Redis a nodo singolo (API ElastiCache)

La procedura seguente descrive come eseguire un dimensionamento verticale di un cluster di cache Redis a nodo singolo utilizzando l'API ElastiCache.

Per ridurre un cluster di cache Redis a nodo singolo (API ElastiCache)

1. Stabilire per quali tipi di nodo è possibile eseguire la riduzione verticale eseguendo l'operazione `ListAllowedNodeTypeModifications` dell'API ElastiCache con il seguente parametro.
 - `CacheClusterId` - Il nome del cluster di cache Redis a nodo singolo che desideri effettuare ridurre verticalmente.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListAllowedNodeTypeModifications  
&CacheClusterId=MyRedisCacheCluster  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta [ListAllowedNodeTypeModifications](#) nella Guida di riferimento delle API di Amazon ElastiCache.

2. Modifica il cluster di cache esistente specificando il cluster di cache da aumentare e il nuovo tipo di nodo più grande, utilizzando l'operazione `ModifyCacheCluster` dell'API ElastiCache e i seguenti parametri.

- `CacheClusterId` : Il nome del cluster di cache che stai dimensionando verso il basso.
- `CacheNodeType` : Il nuovo tipo di nodo più grande a cui desideri dimensionare verso il basso il cluster di cache. Questo valore deve essere uno dei tipi di nodo restituiti dall'operazione `ListAllowedNodeTypeModifications` nella fase 1.
- `CacheParameterGroupName` – [Facoltativo] Utilizzare questo parametro se si usa `reserved-memory` per gestire la memoria prenotata del cluster. Specificare un gruppo di parametri di cache personalizzato che riserva la quantità di memoria corretta per il nuovo tipo di nodo. Se si sta utilizzando `reserved-memory-percent` è possibile omettere questo parametro.
- `ApplyImmediately` : Imposta su `true` per eseguire immediatamente il processo di dimensionamento verso il basso. Per posporre il processo di dimensionamento alla finestra di manutenzione successiva del cluster, utilizzare `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyCacheCluster  
  &ApplyImmediately=true  
  &CacheClusterId=MyRedisCacheCluster  
  &CacheNodeType=cache.m3.xlarge  
  &CacheParameterGroupName=redis32-m2-x1  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta [ModifyCacheCluster](#) nella Guida di riferimento delle API di Amazon ElastiCache.

3. Se è stato utilizzato `ApplyImmediately=true`, controllare lo stato del nuovo cluster di cache utilizzando l'operazione ElastiCache dell'API `DescribeCacheClusters` con il seguente parametro. Quando lo stato cambia in disponibile, è possibile iniziare a utilizzare il nuovo cluster di cache più piccolo.
 - `CacheClusterId` : Il nome del cluster di cache Redis a nodo singolo. Utilizzare questo parametro per descrivere un cluster di cache particolare piuttosto che tutti i cluster di cache.

```
https://elasticache.us-west-2.amazonaws.com/
```

```
?Action=DescribeCacheClusters  
&CacheClusterId=MyRedisCacheCluster  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta [DescribeCacheClusters](#) nella Guida di riferimento delle API di Amazon ElastiCache.

Ridimensionamento dei cluster Redis (modalità cluster disabilitata) con nodi di replica

Un cluster Redis con i nodi di replica (denominato gruppo di replica in API/CLI) offre disponibilità elevata tramite la replica che ha Multi-AZ con funzione di failover automatico abilitata. Un cluster con i nodi di replica è una raccolta logica di un massimo di sei nodi Redis dove un nodo, quello primario, è in grado di fornire le richieste di lettura e scrittura. Tutti gli altri nodi nel cluster sono repliche di sola lettura del nodo primario. I dati scritti sul cluster primario vengono replicati in modo asincrono su tutte le repliche di lettura nel cluster. Poiché Redis (modalità cluster disabilitata) non supporta il partizionamento dei dati su più cluster, ogni nodo in un gruppo di replica Redis (modalità cluster disabilitata) contiene l'intero dataset di cache. I cluster Redis (modalità cluster abilitata) supportano il partizionamento dei dati su un massimo di 500 shard.

Per modificare la capacità dei dati per il cluster, è necessario prima dimensionarlo verso l'alto a un nodo più grande o verso il basso a un nodo più piccolo.

Per modificare la capacità di lettura del cluster, aggiungere più repliche di lettura (fino a un massimo di 5) o rimuoverle.

Il processo di dimensionamento verso l'alto di ElastiCache è progettato per cercare di conservare i dati esistenti e, a tale scopo, richiede una corretta replica Redis. Per i cluster Redis con repliche, consigliamo di programmare una quantità di memoria libera sufficiente per Redis.

Argomenti correlati

- [Alta disponibilità utilizzando gruppi di replica](#)
- [Replica: Redis \(modalità cluster disabilitata\) e Redis \(modalità cluster abilitata\)](#)
- [Riduzione dei tempi di inattività in ElastiCache for Redis con Multi-AZ](#)
- [Verifica della disponibilità di memoria sufficiente per creare uno snapshot Redis](#)

Argomenti

- [Dimensionamento verso l'alto di cluster Redis con repliche](#)
- [Dimensionamento verso il basso delle risorse dei cluster Redis con repliche](#)
- [Aumento della capacità di lettura](#)
- [Diminuzione della capacità di lettura](#)

Dimensionamento verso l'alto di cluster Redis con repliche

Amazon ElastiCache offre console, CLI e support API per dimensionare il gruppo di replica Redis (modalità cluster disabilitata).

Una volta avviato il processo, ElastiCache mostra il seguente messaggio:

1. Avvia un nuovo gruppo di replica utilizzando il nuovo tipo di nodo.
2. Copia tutti i dati da un nodo primario corrente al nuovo nodo primario.
3. Sincronizza le nuove repliche di lettura con il nuovo nodo primario.
4. Aggiorna le voci DNS in modo che puntino ai nuovi nodi. Per questo motivo non è necessario aggiornare gli endpoint nell'applicazione. Per Redis 5.0.5 e versioni successive, puoi ridimensionare cluster abilitati per il failover automatico mentre il cluster continua a rimanere online e servire richieste in entrata. Sulla versione 4.0.10 e precedenti, si potrebbe verificare una breve interruzione di letture e scritture su versioni precedenti dal nodo primario mentre la voce DNS viene aggiornata.
5. Elimina i nodi precedenti (CLI/API: gruppo di replica). Noterai una breve interruzione (pochi secondi) di letture e scritture dai vecchi nodi perché le connessioni ai vecchi nodi verranno disconnesse.

La durata di questo processo dipende dal tipo di nodo e dalla quantità di dati presenti nel cluster.

Come illustrato nella tabella seguente, l'operazione di dimensionamento verso l'alto di Redis si blocca in caso vi sia un upgrade del motore pianificato per la finestra di manutenzione successiva del cluster.

Operazioni Redis bloccate

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

Se hai un'operazione in sospeso che ti blocca, puoi svolgere una delle seguenti operazioni.

- Programmare l'operazione di dimensionamento verso l'alto del motore Redis per una finestra di manutenzione successiva, deselegnando la casella di controllo Apply immediately (Applica immediatamente) (on la CLI utilizzare `--no-apply-immediately`; con l'API, ci si può avvalere di `ApplyImmediately=false`).
- Attendere una finestra di manutenzione successiva per l'operazione di dimensionamento verso l'alto di Redis.
- Aggiungere l'aggiornamento del motore Redis a questa modifica del cluster di cache con la casella di controllo Apply Immediately (Applica immediatamente) selezionata (on la CLI utilizzare `--apply-immediately`; con l'API, ci si può avvalere di `ApplyImmediately=true`). In questo modo si sblocca l'operazione di dimensionamento verso l'alto per eseguire immediatamente l'aggiornamento automatico del motore.

Le sezioni seguenti descrivono come dimensionare verso l'alto il cluster Redis con repliche utilizzando la console ElastiCache, l'AWS CLI e l'API ElastiCache.

Important

Se il gruppo di parametri utilizza `reserved-memory` per riservare memoria per un overhead Redis, prima di iniziare il dimensionamento verificare di disporre di un gruppo di parametri personalizzato che riserva una quantità corretta di memoria per il nuovo tipo di nodo. In alternativa, è possibile modificare un gruppo di parametri personalizzato in modo che utilizzi `reserved-memory-percent` e il gruppo di parametri per il nuovo cluster. Se stai utilizzando `reserved-memory-percent`, questa operazione non è necessaria. Per ulteriori informazioni, consulta [Gestione della memoria prenotata](#).

Dimensionamento verso l'alto di un cluster Redis con repliche (console)

Il tempo necessario per dimensionare l'alto a un tipo di nodo più grande varia a seconda del tipo di nodo e della quantità di dati nel cluster corrente.

La procedura seguente ridimensiona il cluster con le repliche dal tipo di nodo corrente a un nuovo tipo più grande di nodo utilizzando la console ElastiCache. Durante questo processo, si potrebbe verificare una breve interruzione di letture e scritture per altre versioni del nodo primario mentre la voce DNS viene aggiornata. Si potrebbe verificare un tempo di inattività inferiore al secondo per nodi in esecuzione su versioni 5.0.6 e superiore a qualche secondo per versioni precedenti.

Dimensionamento verso l'alto di un cluster con repliche (console)

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dal riquadro di navigazione scegli Cluster Redis.
3. Nell'elenco di cluster, scegliere quello da dimensionare. Questo cluster deve eseguire un motore Redis e non il motore Redis in cluster.
4. Scegli Modifica.
5. Nella procedura guidata Modify Cluster (Modifica cluster):
 - a. Scegliere il tipo di nodo a cui dimensionare dall'elenco Node type (Tipo di nodo). Si noti che non tutti i tipi di nodo sono disponibili per il dimensionamento.
 - b. Se stai utilizzando `reserved-memory` per gestire la tua memoria, dall'elenco Parameter Group (Gruppo di parametri), seleziona il gruppo di parametri personalizzati che riserva la quantità corretta di memoria per il nuovo tipo di nodo.
6. Per eseguire subito il processo di dimensionamento verso l'alto, scegliere la casella Apply immediately (Applica immediatamente). Se non viene selezionata la casella Apply immediately (Applica immediatamente), il processo di dimensionamento verso l'alto viene eseguito nel corso della successiva finestra di manutenzione del cluster.
7. Scegli Modifica.
8. Quando lo stato del cluster cambia da modifica in corso a disponibile, il cluster è stato dimensionato a un nuovo tipo di nodo. Non è necessario aggiornare gli endpoint nell'applicazione.

Aumentare le dimensioni di un gruppo di replica Redis (AWS CLI)

La procedura seguente ridimensiona il gruppo di replica dal tipo di nodo corrente a un nuovo tipo più grande di nodo utilizzando l'AWS CLI. Durante questo processo, ElastiCache for Redis Aggiorna le voci DNS in modo che puntino ai nuovi nodi. Per questo motivo non è necessario aggiornare gli endpoint nell'applicazione. Per Redis 5.0.5 e versioni successive, puoi ridimensionare cluster abilitati per il failover automatico mentre il cluster continua a rimanere online e servire richieste in entrata. Sulla versione 4.0.10 e precedenti, si potrebbe verificare una breve interruzione di letture e scritture su versioni precedenti dal nodo primario mentre la voce DNS viene aggiornata.

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 di cache corrente.

Per aumentare un gruppo di replica Redis (AWS CLI)

1. Stabilire per quali tipi di nodi è possibile eseguire il dimensionamento verso l'alto eseguendo il comando `list-allowed-node-type-modifications` dell'AWS CLI con il seguente parametro.
 - `--replication-group-id` = il nome del gruppo di replica. Utilizzare questo parametro per descrivere un gruppo di replica particolare piuttosto che tutti i gruppi di replica.

Per Linux, macOS o Unix:

```
aws elasticache list-allowed-node-type-modifications \  
  --replication-group-id my-repl-group
```

Per Windows:

```
aws elasticache list-allowed-node-type-modifications ^  
  --replication-group-id my-repl-group
```

L'output di questa operazione è simile al seguente (in formato JSON).

```
{  
  "ScaleUpModifications": [  
    "cache.m3.2xlarge",  
    "cache.m3.large",  
    "cache.m3.xlarge",  
    "cache.m4.10xlarge",  
    "cache.m4.2xlarge",  
    "cache.m4.4xlarge",  
    "cache.m4.large",  
    "cache.m4.xlarge",  
    "cache.r3.2xlarge",  
    "cache.r3.4xlarge",  
    "cache.r3.8xlarge",  
    "cache.r3.large",  
    "cache.r3.xlarge"  
  ]  
}
```

Per ulteriori informazioni, consulta [list-allowed-node-type-modifications](#) nella documentazione di riferimento AWS CLI.

2. Aumenta le dimensioni del gruppo di replica corrente fino al nuovo tipo di nodo utilizzando il comando `modify-replication-group` AWS CLI con i seguenti parametri.
 - `--replication-group-id` : il nome del gruppo di replica.
 - `--cache-node-type` : Il nuovo tipo di nodo più grande di cluster di cache nel gruppo di replica. Questo valore deve essere uno dei tipi di istanza restituiti dal comando `list-allowed-node-type-modifications` nella fase 1.
 - `--cache-parameter-group-name` – [Facoltativo] Utilizzare questo parametro se si usa `reserved-memory` per gestire la memoria prenotata del cluster. Specificare un gruppo di parametri di cache personalizzato che riserva la quantità di memoria corretta per il nuovo tipo di nodo. Se si sta utilizzando `reserved-memory-percent` è possibile omettere questo parametro.
 - `--apply-immediately` : Consente di applicare immediatamente il dimensionamento verso l'alto. Per posporre l'operazione di dimensionamento verso l'alto alla finestra di manutenzione successiva, utilizzare `--no-apply-immediately`.

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id my-repl-group \  
  --cache-node-type cache.m3.xlarge \  
  --cache-parameter-group-name redis32-m3-2x1 \  
  --apply-immediately
```

Per Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id my-repl-group ^  
  --cache-node-type cache.m3.xlarge ^  
  --cache-parameter-group-name redis32-m3-2x1 \  
  --apply-immediately
```

L'output di questo comando è simile al seguente (in formato JSON).

```
{
  "ReplicationGroup": {
    "Status": "available",
    "Description": "Some description",
    "NodeGroups": [{
      "Status": "available",
      "NodeGroupMembers": [{
        "CurrentRole": "primary",
        "PreferredAvailabilityZone": "us-west-2b",
        "CacheNodeId": "0001",
        "ReadEndpoint": {
          "Port": 6379,
          "Address": "my-repl-group-001.8fdx4s.0001.usw2.cache.amazonaws.com"
        }
      }],
      "CacheClusterId": "my-repl-group-001"
    },
    {
      "CurrentRole": "replica",
      "PreferredAvailabilityZone": "us-west-2c",
      "CacheNodeId": "0001",
      "ReadEndpoint": {
        "Port": 6379,
        "Address": "my-repl-group-002.8fdx4s.0001.usw2.cache.amazonaws.com"
      }
    },
    {
      "CacheClusterId": "my-repl-group-002"
    }
  ],
  "NodeGroupId": "0001",
  "PrimaryEndpoint": {
    "Port": 6379,
    "Address": "my-repl-group.8fdx4s.ng.0001.usw2.cache.amazonaws.com"
  }
},
"ReplicationGroupId": "my-repl-group",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "disabled",
"SnapshotWindow": "12:00-13:00",
"SnapshottingClusterId": "my-repl-group-002",
"MemberClusters": [
  "my-repl-group-001",
  "my-repl-group-002"
],
}
```

```
"PendingModifiedValues": {}  
}  
}
```

Per ulteriori informazioni, consulta [modify-replication-group](#) nella documentazione di riferimento AWS CLI.

3. Se è stato utilizzato il parametro `--apply-immediately`, monitorare lo stato del gruppo di replica utilizzando il comando AWS CLI `describe-replication-group` con il seguente parametro. Mentre lo stato è ancora impostato su modifica in corso, si potrebbe verificare un tempo di inattività inferiore al secondo per nodi in esecuzione su versioni 5.0.6 e successive e una breve interruzione di letture e scritture per versioni precedenti dal nodo primario mentre la voce DNS viene aggiornata.
 - `--replication-group-id` = il nome del gruppo di replica. Utilizzare questo parametro per descrivere un gruppo di replica particolare piuttosto che tutti i gruppi di replica.

Per Linux, macOS o Unix:

```
aws elasticache describe-replication-groups \  
  --replication-group-id my-replication-group
```

Per Windows:

```
aws elasticache describe-replication-groups ^  
  --replication-group-id my-replication-group
```

Per ulteriori informazioni, consulta [describe-replication-groups](#) nella documentazione di riferimento dell'AWS CLI.

Ridimensionamento di un gruppo di replica Redis (API ElastiCache)

La procedura seguente ridimensiona il gruppo di replica dal tipo di nodo corrente a un nuovo tipo più grande di nodo utilizzando l'API ElastiCache. Per Redis 5.0.5 e versioni successive, puoi ridimensionare cluster abilitati per il failover automatico mentre il cluster continua a rimanere online e servire richieste in entrata. Sulla versione 4.0.10 e precedenti, si potrebbe verificare una breve interruzione di letture e scritture su versioni precedenti dal nodo primario mentre la voce DNS viene aggiornata.

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 di cache corrente.

Per aumentare un gruppo di replica Redis (API ElastiCache)

1. Stabilire quali tipi di nodo è possibile aumentare mediante l'operazione `ListAllowedNodeTypeModifications` dell'API ElastiCache con il seguente parametro.
 - `ReplicationGroupId` : il nome del gruppo di replica. Utilizzare questo parametro per descrivere un gruppo di replica specifico piuttosto che tutti i gruppi di replica.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ListAllowedNodeTypeModifications  
  &ReplicationGroupId=MyReplGroup  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta [ListAllowedNodeTypeModifications](#) nella Guida di riferimento delle API di Amazon ElastiCache.

2. Aumenta le dimensioni del gruppo di replica corrente fino al nuovo tipo di nodo utilizzando l'operazione API ElastiCache `ModifyRedplicationGroup` e i seguenti parametri.
 - `ReplicationGroupId` : il nome del gruppo di replica.
 - `CacheNodeType` : Il nuovo tipo di nodo più grande di cluster di cache nel gruppo di replica. Questo valore deve essere uno dei tipi di istanza restituiti dall'operazione `ListAllowedNodeTypeModifications` nella fase 1.
 - `CacheParameterGroupName` – [Facoltativo] Utilizzare questo parametro se si usa `reserved-memory` per gestire la memoria prenotata del cluster. Specificare un gruppo di parametri di cache personalizzato che riserva la quantità di memoria corretta per il nuovo tipo di nodo. Se si sta utilizzando `reserved-memory-percent` è possibile omettere questo parametro.
 - `ApplyImmediately` : Impostato su `true` permette di applicare immediatamente il processo di dimensionamento verso il basso. Per posporre il processo di dimensionamento verticale alla successiva finestra di manutenzione, utilizzare `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyReplicationGroup  
&ApplyImmediately=true  
&CacheNodeType=cache.m3.2xlarge  
&CacheParameterGroupName=redis32-m3-2x1  
&ReplicationGroupId=myReplGroup  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&Version=2014-12-01  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

Per ulteriori informazioni, consulta [ModifyReplicationGroup](#) nella Guida di riferimento delle API di Amazon ElastiCache.

3. Se è stato utilizzato `ApplyImmediately=true`, monitorare lo stato del gruppo di replica utilizzando l'operazione `DescribeReplicationGroups` dell'API ElastiCache con i seguenti parametri. Quando lo stato cambia da modifica in corso a disponibile, puoi iniziare a scrivere sul nuovo gruppo di replica ingrandito.
 - `ReplicationGroupId` = il nome del gruppo di replica. Utilizzare questo parametro per descrivere un gruppo di replica particolare piuttosto che tutti i gruppi di replica.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeReplicationGroups  
&ReplicationGroupId=MyReplGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta [DescribeReplicationGroups](#) nella Guida di riferimento delle API di Amazon ElastiCache.

Dimensionamento verso il basso delle risorse dei cluster Redis con repliche

Le sezioni seguenti illustrano come ridurre il numero di risorse di un cluster cache Redis (modalità cluster disabilitata) con i nodi di replica a un tipo di nodo più piccolo. La garanzia che il nuovo tipo di nodo più piccolo sia abbastanza grande da accogliere tutti i dati e la gestione associata è molto importante per il successo. Per ulteriori informazioni, consulta [Verifica della disponibilità di memoria sufficiente per creare uno snapshot Redis](#).

Note

Per i cluster che eseguono il tipo di nodo `r6gd`, è possibile dimensionare solo le dimensioni dei nodi all'interno della famiglia di nodi `r6gd`.

Important

Se il gruppo di parametri utilizza `reserved-memory` per riservare memoria per un overhead Redis, prima di iniziare il dimensionamento verificare di disporre di un gruppo di parametri personalizzato che riserva una quantità corretta di memoria per il nuovo tipo di nodo. In alternativa, è possibile modificare un gruppo di parametri personalizzato in modo che utilizzi `reserved-memory-percent` e il gruppo di parametri per il nuovo cluster.

Se stai utilizzando `reserved-memory-percent`, questa operazione non è necessaria. Per ulteriori informazioni, consulta [Gestione della memoria prenotata](#).

Ridurre le risorse del cluster Redis con repliche (console)

Il processo seguente ridimensiona il cluster Redis con i nodi di replica in un tipo di nodo più piccolo utilizzando la console ElastiCache.

Per ridurre le risorse del cluster Redis con i nodi di replica (console)

1. Verificare che il dimensionamento a un tipo di nodo più piccolo sia adeguato ai dati e all'overhead.
2. Se il gruppo di parametri utilizza `reserved-memory` per riservare memoria per un overhead Redis, è necessario avere la garanzia di disporre di un gruppo di parametri personalizzato per riservare una quantità corretta di memoria per il nuovo tipo di nodo.

- In alternativa, puoi modificare un gruppo di parametri personalizzato per utilizzare `reserved-memory-percent`. Per ulteriori informazioni, consulta [Gestione della memoria prenotata](#).
3. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
 4. Nell'elenco dei cluster, scegliere quello da diminuire. Questo cluster deve eseguire un motore Redis e non il motore Redis in cluster.
 5. Scegli Modifica.
 6. Nella procedura guidata Modify Cluster (Modifica cluster):
 - a. Scegliere il tipo di nodo che si desidera diminuire dall'elenco Node type (Tipo di nodo).
 - b. Se stai utilizzando `reserved-memory` per gestire la tua memoria, dall'elenco Parameter Group (Gruppo di parametri), seleziona il gruppo di parametri personalizzati che riserva la quantità corretta di memoria per il nuovo tipo di nodo.
 7. Per eseguire subito il processo di dimensionamento, scegliere la casella Apply immediately (Applica immediatamente). Se non viene selezionata la casella Apply immediately (Applica immediatamente), il processo di dimensionamento viene eseguito nel corso della successiva finestra di manutenzione del cluster.
 8. Scegli Modifica.
 9. Quando lo stato del cluster cambia da modifica in corso a disponibile, il cluster è stato dimensionato a un nuovo tipo di nodo. Non è necessario aggiornare gli endpoint nell'applicazione.

Ridurre le dimensioni di un gruppo di replica Redis (AWS CLI)

La procedura seguente ridimensiona il gruppo di replica dal tipo di nodo corrente a un nuovo tipo di nodo di minori dimensioni utilizzando l'AWS CLI. Durante questo processo, ElastiCache for Redis Aggiorna le voci DNS in modo che puntino ai nuovi nodi. Per questo motivo non è necessario aggiornare gli endpoint nell'applicazione. Per Redis 5.0.5 e versioni successive, puoi ridimensionare cluster abilitati per il failover automatico mentre il cluster continua a rimanere online e servire richieste in entrata. Sulla versione 4.0.10 e precedenti, si potrebbe verificare una breve interruzione di letture e scritture su versioni precedenti dal nodo primario mentre la voce DNS viene aggiornata.

Tuttavia, le letture dai cluster di cache della replica di lettura continuano ininterrotte.

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 di cache attuale.

Per ridurre un gruppo di replica Redis (AWS CLI)

1. Stabilire per quali tipi di nodi è possibile eseguire il dimensionamento verso il basso eseguendo il comando `list-allowed-node-type-modifications` dell'AWS CLI con il seguente parametro.
 - `--replication-group-id` : il nome del gruppo di replica. Utilizzare questo parametro per descrivere un gruppo di replica particolare piuttosto che tutti i gruppi di replica.

Per Linux, macOS o Unix:

```
aws elasticache list-allowed-node-type-modifications \  
  --replication-group-id my-repl-group
```

Per Windows:

```
aws elasticache list-allowed-node-type-modifications ^  
  --replication-group-id my-repl-group
```

L'output di questa operazione è simile al seguente (in formato JSON).

```
{  
  "ScaleDownModifications": [  
    "cache.m3.2xlarge",  
    "cache.m3.large",  
    "cache.m3.xlarge",  
    "cache.m4.10xlarge",  
    "cache.m4.2xlarge",  
    "cache.m4.4xlarge",  
    "cache.m4.large",  
    "cache.m4.xlarge",  
    "cache.r3.2xlarge",  
    "cache.r3.4xlarge",  
    "cache.r3.8xlarge",  
    "cache.r3.large",  
    "cache.r3.xlarge"  
  ]  
}
```

Per ulteriori informazioni, consulta [list-allowed-node-type-modifications](#) nella documentazione di riferimento AWS CLI.

2. Aumenta le dimensioni del gruppo di replica corrente fino al nuovo tipo di nodo utilizzando il comando `modify-replication-group` AWS CLI con i seguenti parametri.
 - `--replication-group-id` : il nome del gruppo di replica.
 - `--cache-node-type` : Il nuovo tipo di nodo più piccolo da usare nei cluster di cache in questo gruppo di replica. Questo valore deve essere uno dei tipi di istanza restituiti dal comando `list-allowed-node-type-modifications` nella fase 1.
 - `--cache-parameter-group-name` – [Facoltativo] Utilizzare questo parametro se si usa `reserved-memory` per gestire la memoria prenotata del cluster. Specificare un gruppo di parametri di cache personalizzato che riserva la quantità di memoria corretta per il nuovo tipo di nodo. Se si sta utilizzando `reserved-memory-percent` è possibile omettere questo parametro.
 - `--apply-immediately` : Consente di applicare immediatamente il dimensionamento verso l'alto. Per posporre l'operazione di dimensionamento verso l'alto alla finestra di manutenzione successiva, utilizzare `--no-apply-immediately`.

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id my-repl-group \  
  --cache-node-type cache.t2.small \  
  --cache-parameter-group-name redis32-m3-2x1 \  
  --apply-immediately
```

Per Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id my-repl-group ^  
  --cache-node-type cache.t2.small ^  
  --cache-parameter-group-name redis32-m3-2x1 \  
  --apply-immediately
```

L'output di questo comando è simile al seguente (in formato JSON).

```

{"ReplicationGroup": {
  "Status": "available",
  "Description": "Some description",
  "NodeGroups": [
    {
      "Status": "available",
      "NodeGroupMembers": [
        {
          "CurrentRole": "primary",
          "PreferredAvailabilityZone": "us-west-2b",
          "CacheNodeId": "0001",
          "ReadEndpoint": {
            "Port": 6379,
            "Address": "my-repl-
group-001.8fdx4s.0001.usw2.cache.amazonaws.com"
          },
          "CacheClusterId": "my-repl-group-001"
        },
        {
          "CurrentRole": "replica",
          "PreferredAvailabilityZone": "us-west-2c",
          "CacheNodeId": "0001",
          "ReadEndpoint": {
            "Port": 6379,
            "Address": "my-repl-
group-002.8fdx4s.0001.usw2.cache.amazonaws.com"
          },
          "CacheClusterId": "my-repl-group-002"
        }
      ],
      "NodeGroupId": "0001",
      "PrimaryEndpoint": {
        "Port": 6379,
        "Address": "my-repl-
group.8fdx4s.ng.0001.usw2.cache.amazonaws.com"
      }
    }
  ],
  "ReplicationGroupId": "my-repl-group",
  "SnapshotRetentionLimit": 1,
  "AutomaticFailover": "disabled",
  "SnapshotWindow": "12:00-13:00",

```

```
    "SnapshottingClusterId": "my-repl-group-002",
    "MemberClusters": [
      "my-repl-group-001",
      "my-repl-group-002",
    ],
    "PendingModifiedValues": {}
  }
}
```

Per ulteriori informazioni, consulta [modify-replication-group](#) nella documentazione di riferimento AWS CLI.

3. Se è stato utilizzato il parametro `--apply-immediately`, monitorare lo stato del gruppo di replica utilizzando il comando AWS CLI `describe-replication-group` con il seguente parametro. Quando lo stato cambia da modifica in corso a disponibile, puoi iniziare a scrivere sul nuovo gruppo di replica ridotto.
 - `--replication-group-id`: il nome del gruppo di replica. Utilizzare questo parametro per descrivere un gruppo di replica particolare piuttosto che tutti i gruppi di replica.

Per Linux, macOS o Unix:

```
aws elasticache describe-replication-group \
  --replication-group-id my-replication-group
```

Per Windows:

```
aws elasticache describe-replication-groups ^
  --replication-group-id my-replication-group
```

Per ulteriori informazioni, consulta [describe-replication-groups](#) nella documentazione di riferimento dell'AWS CLI.

Ridurre le dimensioni di un gruppo di replica Redis (API ElastiCache)

La procedura seguente ridimensiona il gruppo di replica dal tipo di nodo corrente a un nuovo tipo di nodo di minori dimensioni utilizzando l'API ElastiCache. Durante questo processo, ElastiCache for Redis Aggiorna le voci DNS in modo che puntino ai nuovi nodi. Per questo motivo non è necessario aggiornare gli endpoint nell'applicazione. Per Redis 5.0.5 e versioni successive, puoi ridimensionare

cluster abilitati per il failover automatico mentre il cluster continua a rimanere online e servire richieste in entrata. Sulla versione 4.0.10 e precedenti, si potrebbe verificare una breve interruzione di letture e scritture su versioni precedenti dal nodo primario mentre la voce DNS viene aggiornata. Tuttavia, le letture dai cluster di cache della replica di lettura continuano ininterrotte.

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 di cache attuale.

Per eseguire la riduzione verticale di un gruppo di replica Redis (API ElastiCache)

1. Stabilire quali tipi di nodo è possibile diminuire mediante l'operazione `ListAllowedNodeTypeModifications` dell'API ElastiCache con il seguente parametro.

- `ReplicationGroupId` = il nome del gruppo di replica. Utilizzare questo parametro per descrivere un gruppo di replica specifico piuttosto che tutti i gruppi di replica.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListAllowedNodeTypeModifications  
&ReplicationGroupId=MyReplGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta [ListAllowedNodeTypeModifications](#) nella Guida di riferimento delle API di Amazon ElastiCache.

2. Aumenta le dimensioni del gruppo di replica corrente fino al nuovo tipo di nodo utilizzando l'operazione API ElastiCache `ModifyRedplicationGroup` e i seguenti parametri.

- `ReplicationGroupId` : il nome del gruppo di replica.
- `CacheNodeType` Il nuovo tipo di nodo più piccolo da usare nei cluster di cache in questo gruppo di replica. Questo valore deve essere uno dei tipi di istanza restituiti dall'operazione `ListAllowedNodeTypeModifications` nella fase 1.
- `CacheParameterGroupName` – [Facoltativo] Utilizzare questo parametro se si usa `reserved-memory` per gestire la memoria prenotata del cluster. Specificare un gruppo di parametri di cache personalizzato che riserva la quantità di memoria corretta per il nuovo

tipo di nodo. Se si sta utilizzando `reserved-memory-percent` è possibile omettere questo parametro.

- `ApplyImmediately` - Impostato su `true` permette di applicare immediatamente il processo di dimensionamento verso il basso. Per posporre il processo di dimensionamento verso il basso alla successiva finestra di manutenzione, utilizzare `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyReplicationGroup  
  &ApplyImmediately=true  
  &CacheNodeType=cache.m3.2xlarge  
  &CacheParameterGroupName=redis32-m3-2x1  
  &ReplicationGroupId=myReplGroup  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &Version=2014-12-01  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Per ulteriori informazioni, consulta [ModifyReplicationGroup](#) nella Guida di riferimento delle API di Amazon ElastiCache.

3. Se è stato utilizzato `ApplyImmediately=true`, monitorare lo stato del gruppo di replica utilizzando l'operazione `DescribeReplicationGroups` dell'API ElastiCache con i seguenti parametri. Quando lo stato cambia da modifica in corso a disponibile, puoi iniziare a scrivere sul nuovo gruppo di replica ridotto.
 - `ReplicationGroupId` : il nome del gruppo di replica. Utilizzare questo parametro per descrivere un gruppo di replica particolare piuttosto che tutti i gruppi di replica.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeReplicationGroups  
  &ReplicationGroupId=MyReplGroup  
  &Version=2015-02-02  
  &SignatureVersion=4
```

```
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta [DescribeReplicationGroups](#) nella Guida di riferimento delle API di Amazon ElastiCache.

Aumento della capacità di lettura

Per aumentare la capacità di lettura, aggiungere le repliche di lettura (fino a un massimo di cinque) al gruppo di replica Redis.

Puoi dimensionare la capacità di lettura del cluster Redis utilizzando la console , la AWS CLI o l'API ElastiCache. Per ulteriori informazioni, consulta [Aggiunta di una replica di lettura per i gruppi di replica Redis \(modalità cluster disabilitata\)](#).

Diminuzione della capacità di lettura

Per diminuire la capacità di lettura, eliminare una o più repliche di lettura dal cluster Redis con le repliche (chiamate gruppo di replica nell'API/CLI). Se il cluster ha abilitata la funzione Multi-AZ con failover automatico, non puoi eliminare l'ultima replica di lettura senza prima disabilitare la funzione Multi-AZ. Per ulteriori informazioni, consulta [Modifica di un gruppo di replica](#).

Per ulteriori informazioni, consulta [Eliminazione di una replica di lettura, per i gruppi di replica Redis \(modalità cluster disabilitata\)](#).

Ridimensionamento dei cluster in Redis (modalità cluster attivata)

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 Redis (modalità cluster attivata). 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 stabilire se i nodi sono sottoposti a utilizzo elevato di memoria monitorando i parametri seguenti: `FreeableMemory`, `SwapUsage` e `BytesUseForCache`.

- 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 di throughput monitorando i seguenti parametri: `CPUUtilization`, `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.

È possibile monitorare l'uso del cluster per stabilire se ridurre le dimensioni in modo sicuro utilizzando i seguenti parametri: `FreeableMemory`, `SwapUsage`, `BytesUseForCache`, `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 ridimensionare un cluster Redis (modalità cluster abilitata): ridimensionamento orizzontale e verticale.

- Il ridimensionamento orizzontale permette di modificare il numero di gruppi di nodi (shard) nel gruppo di replica aggiungendo o rimuovendo gruppi di nodi (shard). Il processo di resharding online consente il ridimensionamento in entrambe le direzioni mentre il cluster continua a servire le richieste in arrivo.

Configurare gli slot nel nuovo cluster in modo diverso rispetto a quando si trovavano nel vecchio cluster. Solo metodo offline.

- 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.

Per ulteriori informazioni, consulta [Select cache node size](#).


Indice

- [Reshard offline e ribilanciamento dello shard per Redis \(modalità cluster abilitata\)](#)
- [Reshard online e ribilanciamento della partizione per Redis \(modalità cluster abilitata\)](#)
 - [Aggiunta delle partizioni con il resharding online](#)
 - [Rimozione delle partizioni con il resharding online](#)
 - [Rimozione delle partizioni \(Console\)](#)
 - [Rimozione delle partizioni \(AWS CLI\)](#)
 - [Rimozione degli shard \(API\) ElastiCache](#)
 - [Ribilanciamento delle partizioni online](#)
 - [Ribilanciamento delle partizioni online \(Console\)](#)
 - [Ribilanciamento delle partizioni online \(AWS CLI\)](#)
 - [Ribilanciamento degli shard online \(API\) ElastiCache](#)

- [Ridimensionamento verticale online tramite la modifica del tipo di nodo](#)
 - [Dimensionamento verso l'alto online](#)
 - [Dimensionamento verso l'alto di cluster di cache Redis \(Console\)](#)
 - [Dimensionamento verso l'alto di cluster di cache Redis \(AWS CLI\)](#)
 - [Ridimensionamento dei cluster di cache Redis \(API ElastiCache\)](#)
 - [Dimensionamento verso il basso online](#)
 - [Dimensionamento verso il basso di cluster di cache Redis \(Console\)](#)
 - [Dimensionamento verso il basso di cluster di cache Redis \(AWS CLI\)](#)
 - [Ridimensionamento dei cluster di cache Redis \(API ElastiCache\)](#)

Reshard offline e ribilanciamento dello shard per Redis (modalità cluster abilitata)

Il vantaggio principale derivante dalla riconfigurazione degli shard offline è la possibilità di andare oltre la semplice aggiunta o rimozione degli shard dal gruppo di replica. Durante il resharding offline, oltre a modificare il numero di shard nel gruppo di replica, è possibile:

 Note

Il resharding non in linea non è supportato sui cluster Redis con il tiering di dati abilitato. Per ulteriori informazioni, consulta [Tiering di dati](#).

- Cambiare il tipo di nodo per il gruppo di replica.
- Specificare la zona di disponibilità per ogni nodo nel gruppo di replica.
- Effettuare l'upgrade a una versione del motore più recente.
- Specificare il numero dei nodi di replica in ogni shard in modo indipendente.
- Specificare il keyspace per ogni shard.

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 le shard Redis (modalità cluster abilitata) non in linea

1. Creare un backup manuale del cluster Redis esistente. Per ulteriori informazioni, consulta [Esecuzione di backup manuali](#).
2. Creare un nuovo cluster eseguendo il ripristino dal backup. Per ulteriori informazioni, consulta [Ripristino da un backup in una nuova cache](#).
3. Aggiornare gli endpoint nell'applicazione agli endpoint del nuovo cluster. Per ulteriori informazioni, consulta [Individuazione degli endpoint di connessione](#).

Reshard online e ribilanciamento della partizione per Redis (modalità cluster abilitata)

Utilizzando il resharding online e il ribilanciamento degli shard con ElastiCache Amazon for Redis versione 3.2.10 o successiva, puoi scalare il ElastiCache tuo for Redis (modalità cluster abilitata) 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:

- Dimensionamento orizzontale : Aumenta la capacità in lettura e in scrittura aggiungendo partizioni (gruppi di nodi) al cluster Redis (modalità cluster abilitata) (gruppo di replica).

Se si aggiungono una o più partizioni al gruppo di replica, il numero di nodi nella nuova partizione è lo stesso del numero di nodi nella più piccola delle partizioni esistenti.

- Riduzione orizzontale : riduzione della capacità di lettura e scrittura, e quindi dei costi, rimuovendo frammenti dal cluster Redis (modalità cluster abilitata).
- Ribilanciamento: sposta gli spazi chiave tra gli shard del tuo cluster ElastiCache for Redis (modalità cluster abilitata) in modo che siano distribuiti il più equamente possibile tra gli shard.

Non puoi eseguire le operazioni indicate di seguito:

- Configurare le partizioni in modo indipendente:

Non è possibile specificare il keyspace per le partizioni in modo indipendente. Per fare ciò, è necessario utilizzare il processo offline.

Attualmente, le seguenti limitazioni si applicano al resharding e al ribilanciamento online ElastiCache di Redis:

- Questi processi richiedono il motore Redis versione 3.2.10 o successiva. Per ulteriori informazioni sull'aggiornamento della versione del motore, consultare [Versioni e aggiornamenti del motore](#).
- 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 tutte le nuove partizioni è uguale al numero di nodi nella partizione più piccola esistente.
- Durante l'aumento delle dimensioni, tutti i tag comuni a tutte le partizioni esistenti vengono copiati sulle nuove partizioni.
- Quando si esegue la scalabilità orizzontale di un cluster Global Data Store, ElastiCache le funzioni non vengono replicate automaticamente da uno dei nodi esistenti ai nuovi nodi. Consigliamo di caricare le funzioni nelle nuove partizioni dopo aver aumentato orizzontalmente il cluster in modo che tutte le partizioni abbiano le stesse funzioni.

Note

ElastiCache A differenza della versione 7 e successive di Redis: durante la scalabilità orizzontale del cluster, ElastiCache replicherà automaticamente le funzioni caricate in uno dei nodi esistenti (selezionati a caso) sui nuovi nodi. Se la tua applicazione utilizza [Redis Functions](#), ti consigliamo di caricare tutte le funzioni su tutti gli shard prima di eseguire la scalabilità orizzontale in modo che il cluster ElastiCache for Redis non finisca con definizioni di funzioni diverse su shard diversi.

Per ulteriori informazioni, consulta [Ridimensionamento di cluster online](#).

Puoi scalare o ribilanciare orizzontalmente i tuoi cluster ElastiCache for Redis (abilitata in modalità cluster) utilizzando, the e l'API. AWS Management Console AWS CLI ElastiCache

Aggiunta delle partizioni con il resharding online

Puoi aggiungere shard al tuo cluster Redis (abilitata in modalità cluster) utilizzando l'API, o. AWS Management Console AWS CLI ElastiCache Quando aggiungi partizioni a un cluster Redis (modalità cluster abilitata), tutti i tag sulle partizioni esistenti vengono copiati sulle nuove partizioni.

Aggiunta delle partizioni (Console)

Puoi usare il AWS Management Console per aggiungere uno o più shard al tuo cluster Redis (modalità cluster abilitata). Questo processo viene descritto di seguito.

Per aggiungere partizioni al tuo cluster Redis (modalità cluster abilitata)

1. [Apri la ElastiCache console all'indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Dal riquadro di navigazione scegli Cluster Redis.
3. Individuare e scegliere il nome, e non la casella a sinistra del nome del cluster, del cluster Redis (modalità cluster abilitata) a cui aggiungere le partizioni.

Tip

Mostra Redis (modalità cluster abilitata) Redis in cluster nella Modale column

4. Scegliere Add shard (Aggiungere partizioni).
 - a. Per Number of Shards to be added (Numero delle partizioni da aggiungere), scegliere il numero delle partizioni che si desidera aggiungere a questo cluster.
 - b. Per Availability Zone(s) (Zone di disponibilità), scegliere No preference (Nessuna preferenza) o Specify availability zones (Specifica zone di disponibilità).
 - c. Se è stato selezionato Specify availability zones (Specifica zone di disponibilità) per ciascun nodo in ogni partizione seleziona una zona di disponibilità del nodo dall'elenco delle zone di disponibilità.
 - d. Scegli Aggiungi.

Aggiunta delle partizioni (AWS CLI)

La procedura seguente descrive come riconfigurare le partizioni nel cluster Redis (modalità cluster abilitata) aggiungendo le partizioni che utilizzano la AWS CLI.

Utilizzare i seguenti parametri con `modify-replication-group-shard-configuration`.

Parametri

- `--apply-immediately` : Obbligatorio Specifica che l'operazione di riconfigurazione delle partizioni deve essere avviata immediatamente.
- `--replication-group-id` : Obbligatorio Specifica su quale gruppo di replica (cluster) deve essere eseguita l'operazione di riconfigurazione delle partizioni.
- `--node-group-count` : Obbligatorio Specifica il numero delle partizioni (gruppi di nodi) che deve esistere al completamento dell'operazione. Quando si aggiungono le partizioni, il valore di `--node-group-count` deve essere maggiore del numero corrente delle partizioni.

Facoltativamente, è possibile specificare la zona di disponibilità per ogni nodo nel gruppo di replica, utilizzando `--resharding-configuration`.

- `--resharding-configuration` : Opzionale. Un elenco di zone di disponibilità preferite per ogni nodo in ogni partizione nel gruppo di replica. Utilizzare questo parametro solo se il valore di `--node-group-count` è maggiore del numero corrente delle partizioni. Se questo parametro viene omissso durante l'aggiunta di shard, Amazon ElastiCache seleziona le zone di disponibilità per i nuovi nodi.

L'esempio seguente mostra come riconfigurare gli keyspace su quattro partizioni nel cluster Redis (modalità cluster abilitata) `my-cluster`. Specifica inoltre le zone di disponibilità per ogni nodo in ogni partizione. L'operazione inizia immediatamente.

Example - Aggiunta delle partizioni

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group-shard-configuration \  
  --replication-group-id my-cluster \  
  --node-group-count 4 \  
  --resharding-configuration \  
    "PreferredAvailabilityZones=us-east-2a,us-east-2c" \  
    "PreferredAvailabilityZones=us-east-2b,us-east-2a" \  
    "PreferredAvailabilityZones=us-east-2c,us-east-2d" \  
    "PreferredAvailabilityZones=us-east-2d,us-east-2c" \  
  --apply-immediately
```

Per Windows:

```
aws elasticache modify-replication-group-shard-configuration ^
```

```
--replication-group-id my-cluster ^
--node-group-count 4 ^
--resharding-configuration ^
    "PreferredAvailabilityZones=us-east-2a,us-east-2c" ^
    "PreferredAvailabilityZones=us-east-2b,us-east-2a" ^
    "PreferredAvailabilityZones=us-east-2c,us-east-2d" ^
    "PreferredAvailabilityZones=us-east-2d,us-east-2c" ^
--apply-immediately
```

Per ulteriori informazioni, consulta [modify-replication-group-shard-configuration](#) nella documentazione. AWS CLI

Aggiungere frammenti (API) ElastiCache

Puoi utilizzare l' ElastiCache API per riconfigurare gli shard nel tuo cluster Redis (modalità cluster abilitata) online utilizzando l'operazione. `ModifyReplicationGroupShardConfiguration`

Utilizzare i seguenti parametri con `ModifyReplicationGroupShardConfiguration`.

Parametri

- `ApplyImmediately=true` : Obbligatorio Specifica che l'operazione di riconfigurazione delle partizioni deve essere avviata immediatamente.
- `ReplicationGroupId` : Obbligatorio Specifica su quale gruppo di replica (cluster) deve essere eseguita l'operazione di riconfigurazione delle partizioni.
- `NodeGroupCount` : Obbligatorio Specifica il numero delle partizioni (gruppi di nodi) che deve esistere al completamento dell'operazione. Quando si aggiungono le partizioni, il valore di `NodeGroupCount` deve essere maggiore del numero corrente delle partizioni.

Facoltativamente, è possibile specificare la zona di disponibilità per ogni nodo nel gruppo di replica, utilizzando `ReshardingConfiguration`.

- `ReshardingConfiguration` : Opzionale. Un elenco di zone di disponibilità preferite per ogni nodo in ogni partizione nel gruppo di replica. Utilizzare questo parametro solo se il valore di `NodeGroupCount` è maggiore del numero corrente delle partizioni. Se questo parametro viene omissso durante l'aggiunta di shard, Amazon ElastiCache seleziona le zone di disponibilità per i nuovi nodi.

Il processo seguente descrive come riconfigurare gli shard nel cluster Redis (modalità cluster abilitata) aggiungendo shard utilizzando l'API. ElastiCache

Example - Aggiunta dle partizioni

L'esempio seguente mostra come aggiungere due gruppi di nodi al cluster Redis (modalità cluster abilitata) `my-cluster` quindi, al completamento dell'operazione, vi è un totale di quattro gruppi di nodi. Specifica inoltre le zone di disponibilità per ogni nodo in ognle partizioni. L'operazione inizia immediatamente.

```
https://elasticache.us-east-2.amazonaws.com/  
  ?Action=ModifyReplicationGroupShardConfiguration  
  &ApplyImmediately=true  
  &NodeGroupCount=4  
  &ReplicationGroupId=my-cluster  
  
  &ReshardingConfiguration.ReshardingConfiguration.1.PreferredAvailabilityZones.AvailabilityZone  
east-2a  
  
  &ReshardingConfiguration.ReshardingConfiguration.1.PreferredAvailabilityZones.AvailabilityZone  
east-2c  
  
  &ReshardingConfiguration.ReshardingConfiguration.2.PreferredAvailabilityZones.AvailabilityZone  
east-2b  
  
  &ReshardingConfiguration.ReshardingConfiguration.2.PreferredAvailabilityZones.AvailabilityZone  
east-2a  
  
  &ReshardingConfiguration.ReshardingConfiguration.3.PreferredAvailabilityZones.AvailabilityZone  
east-2c  
  
  &ReshardingConfiguration.ReshardingConfiguration.3.PreferredAvailabilityZones.AvailabilityZone  
east-2d  
  
  &ReshardingConfiguration.ReshardingConfiguration.4.PreferredAvailabilityZones.AvailabilityZone  
east-2d  
  
  &ReshardingConfiguration.ReshardingConfiguration.4.PreferredAvailabilityZones.AvailabilityZone  
east-2c  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20171002T192317Z  
  &X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta l'API Reference [ModifyReplicationGroupShardConfiguration](#).
ElastiCache

Rimozione delle partizioni con il resharding online

Puoi rimuovere gli shard dal tuo cluster Redis (modalità cluster abilitata) utilizzando l'API AWS Management Console AWS CLI, oElastiCache .

Argomenti

- [Rimozione delle partizioni \(Console\)](#)
- [Rimozione delle partizioni \(AWS CLI\)](#)
- [Rimozione degli shard \(API\) ElastiCache](#)

Rimozione delle partizioni (Console)

La procedura seguente descrive come riconfigurare le partizioni nel cluster Redis (modalità cluster abilitata) aggiungendo le partizioni che utilizzano la AWS Management Console.

Prima di rimuovere i gruppi di nodi (shard) dal gruppo di replica, ElastiCache assicuratevi che tutti i dati rientrino negli shard rimanenti. Se i dati sono adatti, le partizioni specificate vengono eliminate dal gruppo di replica come richiesto. Se i dati non si adattano ai restanti gruppi di nodi, il processo viene terminato e il gruppo di replica viene lasciato con la stessa configurazione del gruppo di nodi precedente alla richiesta.

È possibile utilizzare il AWS Management Console per rimuovere uno o più shard dal cluster Redis (modalità cluster abilitata). Non puoi rimuovere tutte le partizioni in un gruppo di replica. Invece, è necessario eliminare il gruppo di replica. Per ulteriori informazioni, consulta [Eliminazione di un gruppo di replica](#). La procedura seguente descrive l'eliminazione di una o più partizioni.

Per rimuovere partizioni dal tuo cluster Redis (modalità cluster abilitata)

1. [Apri la ElastiCache console all'indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Dal riquadro di navigazione scegli Cluster Redis.
3. Individuare e scegliere il nome, e non la casella a sinistra del nome del cluster, del cluster Redis (modalità cluster abilitata) da cui rimuovere le partizioni.


 Tip

I cluster Redis (modalità cluster abilitata) hanno un valore di 1 o superiore nella colonna Shard.

4. Dall'elenco delle partizioni, scegliere la casella a sinistra del nome di ogni partizione da eliminare.
5. Scegliere Delete shard (Elimina partizione).

Rimozione delle partizioni (AWS CLI)

La procedura seguente descrive come riconfigurare le partizioni nel cluster Redis (modalità cluster abilitata) aggiungendo le partizioni che utilizzano la AWS CLI.

 Important

Prima di rimuovere i gruppi di nodi (shard) dal gruppo di replica, ElastiCache assicuratevi che tutti i dati rientrino negli shard rimanenti. Se i dati sono adatti, le partizioni specificate (`--node-groups-to-remove`) vengono eliminate dal gruppo di replica come richiesto e i loro spazi chiave vengono mappati nelle partizioni rimanenti. Se i dati non si adattano ai restanti gruppi di nodi, il processo viene terminato e il gruppo di replica viene lasciato con la stessa configurazione del gruppo di nodi precedente alla richiesta.

È possibile utilizzare il AWS CLI per rimuovere uno o più shard dal cluster Redis (modalità cluster abilitata). Non puoi rimuovere tutte le partizioni in un gruppo di replica. Invece, è necessario eliminare il gruppo di replica. Per ulteriori informazioni, consulta [Eliminazione di un gruppo di replica](#).

Utilizzare i seguenti parametri con `modify-replication-group-shard-configuration`.

Parametri

- `--apply-immediately` : Obbligatorio Specifica che l'operazione di riconfigurazione delle partizioni deve essere avviata immediatamente.
- `--replication-group-id` : Obbligatorio Specifica su quale gruppo di replica (cluster) deve essere eseguita l'operazione di riconfigurazione delle partizioni.

- `--node-group-count` : Obbligatorio Specifica il numero delle partizioni (gruppi di nodi) che deve esistere al completamento dell'operazione. Quando si rimuovono le partizioni, il valore di `--node-group-count` deve essere inferiore al numero corrente delle partizioni.
- `--node-groups-to-remove` : Obbligatorio quando `--node-group-count` è inferiore al numero corrente di gruppi di nodi (partizioni). Un elenco di ID partizioni (gruppo di nodi) da rimuovere dal gruppo di replica.

La procedura seguente descrive l'eliminazione di una o più partizioni.

Example - Rimozione delle partizioni

L'esempio seguente mostra come rimuovere due gruppi di nodi dal cluster Redis (modalità cluster abilitata) `my-cluster` quindi, al completamento dell'operazione, vi è un totale di due gruppi di nodi. Gli spazi chiave delle partizioni rimossi sono distribuiti uniformemente tra le partizioni rimanenti.

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group-shard-configuration \  
  --replication-group-id my-cluster \  
  --node-group-count 2 \  
  --node-groups-to-remove "0002" "0003" \  
  --apply-immediately
```

Per Windows:

```
aws elasticache modify-replication-group-shard-configuration ^  
  --replication-group-id my-cluster ^  
  --node-group-count 2 ^  
  --node-groups-to-remove "0002" "0003" ^  
  --apply-immediately
```

Rimozione degli shard (API) ElastiCache

Puoi utilizzare l' ElastiCache API per riconfigurare gli shard nel tuo cluster Redis (modalità cluster abilitata) online utilizzando l'operazione. `ModifyReplicationGroupShardConfiguration`

Il processo seguente descrive come riconfigurare gli shard nel cluster Redis (modalità cluster abilitata) rimuovendo gli shard utilizzando l'API. ElastiCache

⚠ Important

Prima di rimuovere i gruppi di nodi (shard) dal gruppo di replica, ElastiCache assicura che tutti i dati rientrino negli shard rimanenti. Se i dati sono adatti, le partizioni specificate (NodeGroupsToRemove) vengono eliminate dal gruppo di replica come richiesto e i loro spazi chiave vengono mappati nelle partizioni rimanenti. Se i dati non si adattano ai restanti gruppi di nodi, il processo viene terminato e il gruppo di replica viene lasciato con la stessa configurazione del gruppo di nodi precedente alla richiesta.

Puoi utilizzare l' ElastiCache API per rimuovere uno o più shard dal tuo cluster Redis (modalità cluster abilitata). Non puoi rimuovere tutte le partizioni in un gruppo di replica. Invece, è necessario eliminare il gruppo di replica. Per ulteriori informazioni, consulta [Eliminazione di un gruppo di replica](#).

Utilizzare i seguenti parametri con `ModifyReplicationGroupShardConfiguration`.

Parametri

- `ApplyImmediately=true` : Obbligatorio Specifica che l'operazione di riconfigurazione delle partizioni deve essere avviata immediatamente.
- `ReplicationGroupId` : Obbligatorio Specifica su quale gruppo di replica (cluster) deve essere eseguita l'operazione di riconfigurazione delle partizioni.
- `NodeGroupCount` : Obbligatorio Specifica il numero delle partizioni (gruppi di nodi) che deve esistere al completamento dell'operazione. Quando si rimuovono le partizioni, il valore di `NodeGroupCount` deve essere inferiore al numero corrente delle partizioni.
- `NodeGroupsToRemove` : Obbligatorio quando `--node-group-count` è inferiore al numero corrente di gruppi di nodi (partizioni). Un elenco di ID partizioni (gruppo di nodi) da rimuovere dal gruppo di replica.

La procedura seguente descrive l'eliminazione di una o più partizioni.

Example - Rimozione delle partizioni

L'esempio seguente mostra come rimuovere due gruppi di nodi dal cluster Redis (modalità cluster abilitata) `my-cluster` quindi, al completamento dell'operazione, vi è un totale di due gruppi di nodi. Gli spazi chiave dalle partizioni rimossi sono distribuiti uniformemente tra le partizioni rimanenti.

```
https://elasticache.us-east-2.amazonaws.com/
```

```
?Action=ModifyReplicationGroupShardConfiguration
&ApplyImmediately=true
&NodeGroupCount=2
&ReplicationGroupId=my-cluster
&NodeGroupsToRemove.member.1=0002
&NodeGroupsToRemove.member.2=0003
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20171002T192317Z
&X-Amz-Credential=<credential>
```

Ribilanciamento delle partizioni online

Puoi ribilanciare gli shard nel tuo cluster Redis (modalità cluster abilitata) utilizzando l'API, o. AWS Management Console AWS CLI ElastiCache

Argomenti

- [Ribilanciamento delle partizioni online \(Console\)](#)
- [Ribilanciamento delle partizioni online \(AWS CLI\)](#)
- [Ribilanciamento degli shard online \(API\) ElastiCache](#)

Ribilanciamento delle partizioni online (Console)

La procedura seguente descrive come riconfigurare le partizioni nel cluster Redis (modalità cluster abilitata) AWS Management Console mediante un ribilanciamento delle partizioni utilizzando .

Per ribilanciare i keyspace tra le partizioni sul cluster Redis (modalità cluster abilitata)

1. Apri la ElastiCache console all'[indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Dal riquadro di navigazione scegli Cluster Redis.
3. Scegliere il nome, e non la casella a sinistra del nome, del cluster Redis (modalità cluster abilitata) di cui eseguire il ribilanciamento.

Tip

I cluster Redis (modalità cluster abilitata) hanno un valore di 1 o superiore nella colonna Shard.

4. Scegliere Rebalance (Ribilancia).
5. Quando richiesto, scegliere Rebalance (Ribilancia). Potrebbe essere visualizzato un messaggio del tipo: *Gli slot nel gruppo di replica sono distribuiti in maniera uniforme. Nessuna operazione da fare. (Servizio:AmazonElastiCache; Codice di stato: 400; Codice di errore:; ID richiesta: InvalidReplicationGroupState 2246cebd-9721-11e7-8d5b-e1b0f086c8cf)*. Se viene visualizzato il messaggio, scegliere Cancel (Annulla).

Ribilanciamento delle partizioni online (AWS CLI)

Utilizzare i seguenti parametri con `modify-replication-group-shard-configuration`.

Parametri

- `-apply-immediately` : Obbligatorio Specifica che l'operazione di riconfigurazione delle partizioni deve essere avviata immediatamente.
- `--replication-group-id` : Obbligatorio Specifica su quale gruppo di replica (cluster) deve essere eseguita l'operazione di riconfigurazione delle partizioni.
- `--node-group-count` : Obbligatorio Per ribilanciare gli spazi chiave su tutte le partizioni nel cluster, questo valore deve essere lo stesso del numero delle partizioni correnti.

La procedura seguente descrive come riconfigurare le partizioni nel cluster mediante un ribilanciamento delle partizioni utilizzando AWS CLI.

Example - Ribilanciamento delle partizioni in un cluster

L'esempio seguente mostra il ribilanciamento degli slot nel cluster `my-cluster` in modo che gli slot siano distribuiti nel modo più equo possibile. Il valore di `--node-group-count` (4) rappresenta il numero delle partizioni attualmente nel cluster.

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group-shard-configuration \  
  --replication-group-id my-cluster \  
  --node-group-count 4 \  
  --apply-immediately
```

Per Windows:

```
aws elasticache modify-replication-group-shard-configuration ^
  --replication-group-id my-cluster ^
  --node-group-count 4 ^
  --apply-immediately
```

Ribilanciamento degli shard online (API) ElastiCache

Puoi utilizzare l' ElastiCache API per riconfigurare gli shard nel tuo cluster Redis (modalità cluster abilitata) online utilizzando l'operazione. `ModifyReplicationGroupShardConfiguration`

Utilizzare i seguenti parametri con `ModifyReplicationGroupShardConfiguration`.

Parametri

- `ApplyImmediately=true` : Obbligatorio Specifica che l'operazione di riconfigurazione delle partizioni deve essere avviata immediatamente.
- `ReplicationGroupId` : Obbligatorio Specifica su quale gruppo di replica (cluster) deve essere eseguita l'operazione di riconfigurazione delle partizioni.
- `NodeGroupCount` : Obbligatorio Per ribilanciare gli spazi chiave su tutte le partizioni nel cluster, questo valore deve essere lo stesso del numero delle partizioni corrente.

Il processo seguente descrive come riconfigurare gli shard nel cluster Redis (modalità cluster abilitata) ribilanciando gli shard utilizzando l'API. ElastiCache

Example - Ribilanciamento di un cluster

L'esempio seguente mostra il ribilanciamento degli slot nel cluster Redis (modalità cluster abilitata) `my-cluster` in modo che gli slot siano distribuiti nel modo più equo possibile. Il valore di `NodeGroupCount` (4) rappresenta il numero delle partizioni attualmente nel cluster.

```
https://elasticache.us-east-2.amazonaws.com/
  ?Action=ModifyReplicationGroupShardConfiguration
  &ApplyImmediately=true
  &NodeGroupCount=4
  &ReplicationGroupId=my-cluster
  &Version=2015-02-02
  &SignatureVersion=4
  &SignatureMethod=HmacSHA256
  &Timestamp=20171002T192317Z
```

```
&X-Amz-Credential=<credential>
```

Ridimensionamento verticale online tramite la modifica del tipo di nodo

Utilizzando il ridimensionamento verticale con Amazon ElastiCache for Redis versione 3.2.10 o successiva, puoi dimensionare i cluster Redis in modo dinamico senza tempi di inattività. Ciò consente al cluster Redis di soddisfare le richieste anche durante il ridimensionamento.

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 Redis affinché utilizzi un tipo di nodo più grande.

ElastiCache 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, ElastiCache 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:

- Assicurati di disporre di capacità ENI (Elastic Network Interface) sufficiente. In caso di dimensionamento verso il basso, assicurati che il nodo più piccolo disponga di memoria sufficiente per assorbire il traffico previsto.

Per le best practice da seguire per la gestione della memoria, consultare [Gestione della memoria prenotata](#).

- 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.

Indice

- [Dimensionamento verso l'alto online](#)
 - [Dimensionamento verso l'alto di cluster di cache Redis \(Console\)](#)
 - [Dimensionamento verso l'alto di cluster di cache Redis \(AWS CLI\)](#)
 - [Ridimensionamento dei cluster di cache Redis \(API ElastiCache\)](#)
- [Dimensionamento verso il basso online](#)
 - [Dimensionamento verso il basso di cluster di cache Redis \(Console\)](#)
 - [Dimensionamento verso il basso di cluster di cache Redis \(AWS CLI\)](#)
 - [Ridimensionamento dei cluster di cache Redis \(API ElastiCache\)](#)

Dimensionamento verso l'alto online

Argomenti

- [Dimensionamento verso l'alto di cluster di cache Redis \(Console\)](#)
- [Dimensionamento verso l'alto di cluster di cache Redis \(AWS CLI\)](#)
- [Ridimensionamento dei cluster di cache Redis \(API ElastiCache\)](#)

Dimensionamento verso l'alto di cluster di cache Redis (Console)

La procedura seguente descrive come eseguire un dimensionamento di un cluster Redis utilizzando la console di gestione ElastiCache. Durante questo processo, il cluster Redis continuerà ad elaborare le richieste con tempi di inattività minimi.

Per aumentare un cluster Redis (console)

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.

2. Dal riquadro di navigazione scegli Cluster Redis.
3. Nell'elenco dei cluster, scegliere quello da ridimensionare.
4. Scegli Modifica.
5. Nella procedura guidata Modify Cluster (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.
6. Per eseguire subito il processo di dimensionamento verso l'alto, scegliere la casella Apply immediately (Applica immediatamente). Se non viene selezionata la casella Apply immediately (Applica immediatamente), il processo di dimensionamento verso l'alto viene eseguito nel corso della successiva finestra di manutenzione del cluster.
7. Scegli Modifica.

Se, nel passaggio precedente, è stato scelto Apply immediately (Applica immediatamente), lo stato del cluster diventa modifica in corso. Quando lo stato cambia in disponibile, la modifica è completa ed è possibile iniziare a utilizzare il nuovo cluster.

Dimensionamento verso l'alto di cluster di cache Redis (AWS CLI)

La seguente procedura descrive come eseguire un dimensionamento verso l'alto di un cluster di cache Redis utilizzando la AWS CLI. Durante questo processo, il cluster Redis continuerà ad elaborare le richieste con tempi di inattività minimi.

Per l'aumento di un cluster di cache Redis (AWS CLI)

1. Stabilire per quali tipi di nodi è possibile eseguire il dimensionamento verso l'alto eseguendo il comando `list-allowed-node-type-modifications` dell'AWS CLI con il seguente parametro.

Per Linux, macOS o Unix:

```
aws elasticache list-allowed-node-type-modifications \  
  --replication-group-id my-replication-group-id
```

Per Windows:

```
aws elasticache list-allowed-node-type-modifications ^
```

```
--replication-group-id my-replication-group-id
```

L'output del comando in alto è simile al seguente (in formato JSON).

```
{
  "ScaleUpModifications": [
    "cache.m3.2xlarge",
    "cache.m3.large",
    "cache.m3.xlarge",
    "cache.m4.10xlarge",
    "cache.m4.2xlarge",
    "cache.m4.4xlarge",
    "cache.m4.large",
    "cache.m4.xlarge",
    "cache.r3.2xlarge",
    "cache.r3.4xlarge",
    "cache.r3.8xlarge",
    "cache.r3.large",
    "cache.r3.xlarge"
  ],
  "ScaleDownModifications": [
    "cache.t2.micro",
    "cache.t2.small",
    "cache.t2.medium",
    "cache.t1.small"
  ],
}
```

Per ulteriori informazioni, consulta [list-allowed-node-type-modifications](#) nella documentazione di riferimento AWS CLI.

2. Modifica il gruppo di replicare per ridimensionare verso l'alto al nuovo tipo di nodo più grande utilizzando il comando AWS CLI `modify-replication-group` con i seguenti parametri.
 - `--replication-group-id` : Il nome del gruppo di replica che si sta ridimensionando verso l'alto.
 - `--cache-node-type` : Il nuovo tipo di nodo con il quale intendi ridimensionare il cluster di cache. Questo valore deve essere uno dei tipi di nodi restituiti dal comando `list-allowed-node-type-modifications` nella fase 1.
 - `--cache-parameter-group-name` – [Facoltativo] Utilizzare questo parametro se si usa `reserved-memory` per gestire la memoria prenotata del cluster. Specificare un gruppo di

parametri di cache personalizzato che riserva la quantità di memoria corretta per il nuovo tipo di nodo. Se si sta utilizzando `reserved-memory-percent` è possibile omettere questo parametro.

- `--apply-immediately` : Consente di applicare immediatamente il dimensionamento verso l'alto. Per posporre il processo di dimensionamento alla finestra di manutenzione successiva del cluster, utilizzare il parametro `--no-apply-immediately`.

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id my-redis-cluster \  
  --cache-node-type cache.m3.xlarge \  
  --apply-immediately
```

Per Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id my-redis-cluster ^  
  --cache-node-type cache.m3.xlarge ^  
  --apply-immediately
```

L'output del comando in alto è simile al seguente (in formato JSON).

```
{  
  "ReplicationGroup": {  
    "Status": "modifying",  
    "Description": "my-redis-cluster",  
    "NodeGroups": [  
      {  
        "Status": "modifying",  
        "Slots": "0-16383",  
        "NodeGroupId": "0001",  
        "NodeGroupMembers": [  
          {  
            "PreferredAvailabilityZone": "us-east-1f",  
            "CacheNodeId": "0001",  
            "CacheClusterId": "my-redis-cluster-0001-001"  
          },  
          {
```

```

        "PreferredAvailabilityZone": "us-east-1d",
        "CacheNodeId": "0001",
        "CacheClusterId": "my-redis-cluster-0001-002"
    }
  ]
},
"ConfigurationEndpoint": {
  "Port": 6379,
  "Address": "my-redis-
cluster.r7gdfi.clustercfg.us1.cache.amazonaws.com"
},
"ClusterEnabled": true,
"ReplicationGroupId": "my-redis-cluster",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "enabled",
"SnapshotWindow": "07:30-08:30",
"MemberClusters": [
  "my-redis-cluster-0001-001",
  "my-redis-cluster-0001-002"
],
"CacheNodeType": "cache.m3.xlarge",
"DataTiering": "disabled"
"PendingModifiedValues": {}
}
}

```

Per ulteriori informazioni, consulta [modify-replication-group](#) nella documentazione di riferimento AWS CLI.

- Se è stato utilizzato `--apply-immediately`, monitorare lo stato del nuovo cluster di cache utilizzando il comando AWS CLI `describe-cache-clusters` con il seguente parametro. Quando lo stato cambia in disponibile, è possibile iniziare a utilizzare il nuovo cluster di cache più grande.

Ridimensionamento dei cluster di cache Redis (API ElastiCache)

La procedura seguente ridimensiona il cluster di cache dal tipo di nodo corrente a un nuovo tipo di nodo di maggiori dimensioni utilizzando l'API ElastiCache. Durante questo processo, ElastiCache for Redis Aggiorna le voci DNS in modo che puntino ai nuovi nodi. Per questo motivo non è necessario aggiornare gli endpoint nell'applicazione. Per Redis 5.0.5 e versioni successive, puoi ridimensionare cluster abilitati per il failover automatico mentre il cluster continua a rimanere online e servire richieste

in entrata. Sulla versione 4.0.10 e precedenti, si potrebbe verificare una breve interruzione di letture e scritture su versioni precedenti dal nodo primario mentre la voce DNS viene aggiornata.

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 di cache corrente.

Per aumentare un cluster di cache Redis (API ElastiCache)

1. Stabilire quali nodi è possibile aumentare mediante l'operazione `ListAllowedNodeTypeModifications` dell'API ElastiCache con il seguente parametro.

- `ReplicationGroupId` : il nome del gruppo di replica. Utilizzare questo parametro per descrivere un gruppo di replica specifico piuttosto che tutti i gruppi di replica.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListAllowedNodeTypeModifications  
&ReplicationGroupId=MyReplGroup  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta [ListAllowedNodeTypeModifications](#) nella Guida di riferimento delle API di Amazon ElastiCache.

2. Aumenta le dimensioni del gruppo di replica corrente fino al nuovo tipo di nodo utilizzando l'operazione API ElastiCache `ModifyReplicationGroup` e i seguenti parametri.

- `ReplicationGroupId` : il nome del gruppo di replica.
- `CacheNodeType` : Il nuovo tipo di nodo più grande di cluster di cache nel gruppo di replica. Questo valore deve essere uno dei tipi di istanza restituiti dall'operazione `ListAllowedNodeTypeModifications` nella fase 1.
- `CacheParameterGroupName` – [Facoltativo] Utilizzare questo parametro se si usa `reserved-memory` per gestire la memoria prenotata del cluster. Specificare un gruppo di parametri di cache personalizzato che riserva la quantità di memoria corretta per il nuovo tipo di nodo. Se si sta utilizzando `reserved-memory-percent` è possibile omettere questo parametro.

- `ApplyImmediately` : Impostato su `true` permette di applicare immediatamente il processo di dimensionamento verso il basso. Per posporre il processo di dimensionamento verticale alla successiva finestra di manutenzione, utilizzare `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyReplicationGroup  
  &ApplyImmediately=true  
  &CacheNodeType=cache.m3.2xlarge  
  &CacheParameterGroupName=redis32-m3-2x1  
  &ReplicationGroupId=myReplGroup  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &Version=2014-12-01  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Per ulteriori informazioni, consulta [ModifyReplicationGroup](#) nella Guida di riferimento delle API di Amazon ElastiCache.

3. Se è stato utilizzato `ApplyImmediately=true`, monitorare lo stato del gruppo di replica utilizzando l'operazione `DescribeReplicationGroups` dell'API ElastiCache con i seguenti parametri. Quando lo stato cambia da modifica in corso a disponibile, puoi iniziare a scrivere sul nuovo gruppo di replica ingrandito.
- `ReplicationGroupId` = il nome del gruppo di replica. Utilizzare questo parametro per descrivere un gruppo di replica particolare piuttosto che tutti i gruppi di replica.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeReplicationGroups  
  &ReplicationGroupId=MyReplGroup  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z
```

```
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta [DescribeReplicationGroups](#) nella Guida di riferimento delle API di Amazon ElastiCache.

Dimensionamento verso il basso online

Argomenti

- [Dimensionamento verso il basso di cluster di cache Redis \(Console\)](#)
- [Dimensionamento verso il basso di cluster di cache Redis \(AWS CLI\)](#)
- [Ridimensionamento dei cluster di cache Redis \(API ElastiCache\)](#)

Dimensionamento verso il basso di cluster di cache Redis (Console)

La procedura seguente descrive come dimensionare un cluster Redis utilizzando la console di gestione ElastiCache. Durante questo processo, il cluster Redis continuerà ad elaborare le richieste con tempi di inattività minimi.

Per ridurre un cluster Redis (console)

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dal riquadro di navigazione scegli Cluster Redis.
3. Nell'elenco dei cluster, scegliere quello da ridimensionare.
4. Scegli Modifica.
5. Nella procedura guidata Modify Cluster (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.
6. Per eseguire immediatamente il processo di riduzione verticale, scegliere la casella Apply immediately (Applica immediatamente). Se non viene selezionata la casella Apply immediately (Applica immediatamente), il processo di ridimensionamento verso il basso viene eseguito nel corso della successiva finestra di manutenzione del cluster.
7. Scegli Modifica.

Se, nel passaggio precedente, è stato scelto Apply immediately (Applica immediatamente), lo stato del cluster diventa modifica in corso. Quando lo stato cambia in disponibile, la modifica è completa ed è possibile iniziare a utilizzare il nuovo cluster.

Dimensionamento verso il basso di cluster di cache Redis (AWS CLI)

La seguente procedura descrive come eseguire un ridimensionamento verso il basso di un cluster di cache Redis utilizzando la AWS CLI. Durante questo processo, il cluster Redis continuerà ad elaborare le richieste con tempi di inattività minimi.

Per la riduzione di un cluster di cache Redis (AWS CLI)

1. Stabilire per quali tipi di nodi è possibile eseguire il ridimensionamento verso il basso eseguendo il comando AWS CLI `list-allowed-node-type-modifications` con il seguente parametro.

Per Linux, macOS o Unix:

```
aws elasticache list-allowed-node-type-modifications \  
  --replication-group-id my-replication-group-id
```

Per Windows:

```
aws elasticache list-allowed-node-type-modifications ^  
  --replication-group-id my-replication-group-id
```

L'output del comando in alto è simile al seguente (in formato JSON).

```
{  
  "ScaleUpModifications": [  
    "cache.m3.2xlarge",  
    "cache.m3.large",  
    "cache.m3.xlarge",  
    "cache.m4.10xlarge",  
    "cache.m4.2xlarge",  
    "cache.m4.4xlarge",  
    "cache.m4.large",  
    "cache.m4.xlarge",  
    "cache.r3.2xlarge",
```

```
    "cache.r3.4xlarge",
    "cache.r3.8xlarge",
    "cache.r3.large",
    "cache.r3.xlarge"
  ]

  "ScaleDownModifications": [
    "cache.t2.micro",
    "cache.t2.small ",
    "cache.t2.medium ",
    "cache.t1.small"
  ]
}
```

Per ulteriori informazioni, consulta [list-allowed-node-type-modifications](#) nella documentazione di riferimento AWS CLI.

2. Modifica il gruppo di replicare per ridimensionare verso il basso utilizzando il nuovo tipo di nodo di minori dimensioni utilizzando il comando AWS CLI `modify-replication-group` con i seguenti parametri.

- `--replication-group-id` : Il nome del gruppo di replica oggetto della riduzione verticale verso il basso.
- `--cache-node-type` : Il nuovo tipo di nodo con il quale intendi dimensionare il cluster di cache. Questo valore deve essere uno dei tipi di nodi restituiti dal comando `list-allowed-node-type-modifications` nella fase 1.
- `--cache-parameter-group-name` – [Facoltativo] Utilizzare questo parametro se si usa `reserved-memory` per gestire la memoria prenotata del cluster. Specificare un gruppo di parametri di cache personalizzato che riserva la quantità di memoria corretta per il nuovo tipo di nodo. Se si sta utilizzando `reserved-memory-percent` è possibile omettere questo parametro.
- `--apply-immediately` : Consente di applicare immediatamente il dimensionamento verso l'alto. Per posporre il processo di dimensionamento verso il basso alla finestra di manutenzione successiva del cluster, utilizzare il parametro `--no-apply-immediately`.

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
  --replication-group-id my-redis-cluster \
```

```
--cache-node-type cache.t2.micro \  
--apply-immediately
```

Per Windows:

```
aws elasticache modify-replication-group ^  
--replication-group-id my-redis-cluster ^  
--cache-node-type cache.t2.micro ^  
--apply-immediately
```

L'output del comando in alto è simile al seguente (in formato JSON).

```
{  
  "ReplicationGroup": {  
    "Status": "modifying",  
    "Description": "my-redis-cluster",  
    "NodeGroups": [  
      {  
        "Status": "modifying",  
        "Slots": "0-16383",  
        "NodeGroupId": "0001",  
        "NodeGroupMembers": [  
          {  
            "PreferredAvailabilityZone": "us-east-1f",  
            "CacheNodeId": "0001",  
            "CacheClusterId": "my-redis-cluster-0001-001"  
          },  
          {  
            "PreferredAvailabilityZone": "us-east-1d",  
            "CacheNodeId": "0001",  
            "CacheClusterId": "my-redis-cluster-0001-002"  
          }  
        ]  
      }  
    ],  
    "ConfigurationEndpoint": {  
      "Port": 6379,  
      "Address": "my-redis-  
cluster.r7gdfi.clustercfg.use1.cache.amazonaws.com"  
    },  
    "ClusterEnabled": true,  
  }  
}
```



```
"ReplicationGroupId": "my-redis-cluster",
"SnapshotRetentionLimit": 1,
"AutomaticFailover": "enabled",
"SnapshotWindow": "07:30-08:30",
"MemberClusters": [
  "my-redis-cluster-0001-001",
  "my-redis-cluster-0001-002"
],
"CacheNodeType": "cache.t2.micro",
"DataTiering": "disabled"
"PendingModifiedValues": {}
}
}
```

Per ulteriori informazioni, consulta [modify-replication-group](#) nella documentazione di riferimento AWS CLI.

3. Se è stato utilizzato `--apply-immediately`, monitorare lo stato del nuovo cluster di cache utilizzando il comando AWS CLI `describe-cache-clusters` con il seguente parametro. Quando lo stato cambia in disponibile, è possibile iniziare a utilizzare il nuovo cluster di cache più piccolo.

Ridimensionamento dei cluster di cache Redis (API ElastiCache)

La procedura seguente ridimensiona il gruppo di replica dal tipo di nodo corrente a un nuovo tipo di nodo di minori dimensioni utilizzando l'API ElastiCache. Durante questo processo, il cluster Redis continuerà ad elaborare le richieste con tempi di inattività minimi.

Il tempo necessario per ridurre utilizzando un tipo di nodo di minori dimensioni varia a seconda del tipo di nodo e della quantità di dati nel cluster di cache attuale.

Riduzione verticale (API ElastiCache)

1. Stabilire quali nodi è possibile ridurre mediante l'operazione `ListAllowedNodeTypeModifications` dell'API ElastiCache con il seguente parametro.
 - `ReplicationGroupId` = il nome del gruppo di replica. Utilizzare questo parametro per descrivere un gruppo di replica specifico piuttosto che tutti i gruppi di replica.

```
https://elasticache.us-west-2.amazonaws.com/
```

```
?Action=ListAllowedNodeTypeModifications
&ReplicationGroupId=MyReplGroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta [ListAllowedNodeTypeModifications](#) nella Guida di riferimento delle API di Amazon ElastiCache.

2. Dimensionare verticalmente il gruppo di replica corrente affinché utilizzi il nuovo tipo di nodo tramite l'operazione API ElastiCache `ModifyReplicationGroup` con i seguenti parametri.
 - `ReplicationGroupId` - il nome del gruppo di replica.
 - `CacheNodeType` Il nuovo tipo di nodo più piccolo da usare nei cluster di cache in questo gruppo di replica. Questo valore deve essere uno dei tipi di istanza restituiti dall'operazione `ListAllowedNodeTypeModifications` nella fase 1.
 - `CacheParameterGroupName` – [Facoltativo] Utilizzare questo parametro se si usa `reserved-memory` per gestire la memoria prenotata del cluster. Specificare un gruppo di parametri di cache personalizzato che riserva la quantità di memoria corretta per il nuovo tipo di nodo. Se si sta utilizzando `reserved-memory-percent` è possibile omettere questo parametro.
 - `ApplyImmediately` : Impostato su `true` permette di applicare immediatamente il processo di riduzione verticale. Per posporre il processo di dimensionamento verso il basso alla successiva finestra di manutenzione, utilizzare `ApplyImmediately=false`.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyReplicationGroup
&ApplyImmediately=true
&CacheNodeType=cache.t2.micro
&CacheParameterGroupName=redis32-m3-2x1
&ReplicationGroupId=myReplGroup
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&Version=2014-12-01
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
```

```
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

Per ulteriori informazioni, consulta [ModifyReplicationGroup](#) nella Guida di riferimento delle API di Amazon ElastiCache.

Nozioni di base di JSON in ElastiCache per Redis

ElastiCache per Redis supporta il formato JSON (JavaScript Object 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 la serializzarlo e deserializzarlo.

Oltre all'uso delle operazioni dell'API Redis per applicazioni che operano su JSON, ora puoi recuperare e aggiornare in modo efficiente porzioni specifiche di un documento JSON senza necessità di manipolare l'intero oggetto. Ciò può migliorare le prestazioni e ridurre i costi. Puoi cercare anche contenuti del documento JSON utilizzando l'interrogazione JSONPath di [tipo Goessner](#).

Dopo la creazione di un cluster con una versione del motore supportata, il tipo di dati JSON e i comandi associati sono disponibili automaticamente. Questa è l'API compatibile e il RDB compatibile con la versione 2 del modulo RedisJSON, per cui puoi migrare facilmente le applicazioni Redis basate su JSON esistenti in ElastiCache per Redis. Per ulteriori informazioni sui comandi Redis supportati, consulta [Comandi JSON di Redis non supportati](#).

Le metriche correlate a JSON `JsonBasedCmds` e `JsonBasedCmdsLatency` sono incorporate in CloudWatch per monitorare l'utilizzo di questo tipo di dati. Per ulteriori informazioni, consulta la pagina relativa ai [parametri di Redis](#).

Note

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

Argomenti

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

Panoramica del tipo di dati JSON di Redis

ElastiCache per Redis supporta una serie di comandi Redis da utilizzare 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 JSON di Redis.
Valore JSON	Si riferisce a un sottoinsieme di un documento JSON, inclusa la radice che rappresenta l'intero documento. Un valore può essere un container o una voce vuota in un container.
Elemento JSON	Equivalente a un 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 memorizzati internamente in un formato ottimizzato per l'accesso e la modifica rapidi. Questo formato generalmente comporta un consumo di memoria lievemente superiore 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. Puoi controllare la quantità di memoria che consuma un documento JSON utilizzando il comando `JSON.DEBUG MEMORY`.

ACL JSON

- Simile alle categorie per tipo di dati esistenti (`@string`, `@hash` ecc.), viene aggiunta una nuova categoria `@json` per semplificare la gestione dell'accesso a comandi e 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		
JSON.ARRINDEX	y		y		
JSON.ARRINSERT		y	y		
JSON.ARRLEN	y		y		
JSON.ARRPOP		y	y		

Comando JSON	@read	@write	@fast	@slow	@admin
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		
JSON.OBJECTS	y		y		
JSON.OBJECTLEN	y		y		
JSON.RESP	y		y		
JSON.SET		y		y	
JSON.STRAPPEND		y	y		

Comando JSON	@read	@write	@fast	@slow	@admin
JSON.STRL EN	y		y		
JSON.STRL EN	y		y		
JSON.TOGG LE		y	y		
JSON.TYPE	y		y		
JSON.NUMI NCRBY		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. L'argomento path per impostazione predefinita è la radice se è facoltativo e non viene fornito.

Notazione:

- Gli argomenti richiesti sono racchiusi tra parentesi angolari. Esempio: <key>
- Gli argomenti facoltativi sono racchiusi tra parentesi quadre. Esempio: [path]
- Argomenti facoltativi supplementari sono indicati da un'ellissi ("..."). Esempio: [json...]

Sintassi del percorso

JSON Redis supporta due tipi di sintassi del percorso:

- 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 sono sensibili al tipo di sintassi del percorso utilizzata.

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

Sintassi avanzata

Simbolo/espressione	Descrizione
\$	Elemento radice.
. o []	Operatore figlio.
..	Discesa ricorsiva.
*	Carattere jolly. Tutti gli elementi di un oggetto o un array.
[]	Operatore pedice di un array. L'indice è basato su 0.
[,]	Operatore di unione.
[start:end:step]	Operatore di sezionamento di un array.
?()	Applica un'espressione filtro (script) all'array o all'oggetto corrente.

Simbolo/espressione	Descrizione
()	Espressione filtro.
@	Utilizzato in espressioni filtro che fanno riferimento al nodo corrente in elaborazione.
==	Uguale a, utilizzato in espressioni filtro.
!=	Diverso da, utilizzato in espressioni filtro.
>	Maggiore di, utilizzato in espressioni filtro.
>=	Maggiore o uguale a, utilizzato in espressioni filtro.
<	Minore di, utilizzato in espressioni filtro.
<=	Mainore o uguale a, utilizzato in espressioni filtro.
&&	AND logico, utilizzato per combinare più espressioni filtro.
	OR logico, utilizzato per combinare più espressioni filtro.

Examples (Esempi)

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

```
{ "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
  }
}

```

Path	Descrizione
<code>\$.store.book[*].author</code>	Gli autori di tutti i libri del negozio.
<code>\$..author</code>	tutti gli autori.
<code>\$.store.*</code>	Tutti i membri del negozio.
<code>\$["store"].*</code>	Tutti i membri del negozio.

Path	Descrizione
<code>\$.store..price</code>	Il prezzo di ogni cosa 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 al 3 (l'indice finale escluso).
<code>\$.book[0:4:2]</code>	Libri nell'indice 0, 2.
<code>\$.book[?(@.isbn)]</code>	Tutti i libri con un numero ISBN.
<code>\$.book[?(@.price<10)]</code>	Tutti i libri che costano meno di 10 euro.
<code>'\$.book[?(@.price < 10)]'</code>	Tutti i libri che costano meno di 10 euro. (Il percorso deve essere tra virgolette se contiene spazi vuoti.)
<code>'\$.book[?(@["price"] < 10)]'</code>	Tutti i libri che costano meno di 10 euro.
<code>'\$.book[?(@.["price"] < 10)]'</code>	Tutti i libri che costano meno di 10 euro.
<code>\$.book[?(@.price>=10&&@.price<=100)]</code>	Tutti i libri nella fascia di prezzo da 10 a 100 euro, inclusi.
<code>'\$.book[?(@.price>=10 && @.price<=100)]'</code>	Tutti i libri nella fascia di prezzo da 10 a 100 euro, inclusi. (Il percorso deve essere tra virgolette se contiene spazi vuoti.)
<code>\$.book[?(@.sold==true @.in-stock==false)]</code>	Tutti i libri venduti o esauriti.

Path	Descrizione
'\$.book[?(@.sold == true @.in-stock == false)]'	Tutti i libri venduti o esauriti. (Il percorso deve essere tra virgolette se contiene spazi vuoti.)
'\$.store.book[?(@.["category"] == "fiction")]'	Tutti i libri della categoria fiction.
'\$.store.book[?(@.["category"] != "fiction")]'	Tutti i libri appartenenti a categorie nonfiction.

Altri esempi di espressioni 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 figlio.
[]	Operatore pedice di un array. L'indice è basato su 0.

Examples (Esempi)

Path	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 errori comuni..

Prefix	Descrizione
ERR	Un errore generale.
LIMIT	Un errore che si verifica quando vengono superati i limiti delle dimensioni. Ad esempio, il

Prefix	Descrizione
	limite delle dimensioni del documento o il limite di profondità di nidificazione è stato superato.
NONEXISTENT	Una chiave o percorso non esiste.
OUTOFBOUNDARIES	Un indice array esterno ai 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 eseguire interrogazioni di metriche principali, utilizzare il seguente comando Redis:

```
info json_core_metrics
```

Modalità di interazione di ElastiCache per Redis con JSON

La sezione seguente descrive la modalità di interazione di ElastiCache per Redis con i tipi 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. Operazioni tra parentesi vengono eseguite per prime.

Comportamento del limite massimo di nidificazione dei percorsi

Il limite massimo di nidificazione dei percorsi in ElastiCache per Redis è 128. Per cui, un valore come \$.a.b.c.d... può raggiungere solo 128 livelli.

Gestione dei valori numerici

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

Rappresentazioni numeriche:

Quando un numero JSON viene ricevuto nell'input, viene convertito in una delle due rappresentazioni binarie interne: un numero intero con segno a 64 bit o un numero in virgola mobile a doppia precisione IEEE 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.

Comandi aritmetici NUMINCRBY e NUMMULTBY:

- Se entrambi i numeri sono interi e il risultato non rientra nell'intervallo `int64`, diventa automaticamente un numero in 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 doppio IEEE a 64 bit, il comando restituisce un errore `OVERFLOW`.

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

Filtraggio array diretto

ElastiCache per Redis filtra oggetti array direttamente.

Per dati come `[0,1,2,3,4,5,6]` e un'interrogazione percorso come `$[?(@<4)]` oppure dati come `{"my_key": [0,1,2,3,4,5,6]}` e un'interrogazione percorso come `$.my_key[?(@<4)]`, ElastiCache per Redis restituisce `[1,2,3]` in entrambe le circostanze.

Comportamento di indicizzazione array

ElastiCache per Redis consente indici sia positivi che negativi per gli array. Per un array di lunghezza cinque, 0 interroga il primo elemento, 1 il secondo e così via. I numeri negativi iniziano alla fine dell'array, per cui -1 interroga il quinto elemento, -2 il quarto e così via.

Per garantire un comportamento prevedibile per i clienti, ElastiCache per Redis non arrotonda indici array per difetto o per eccesso, per cui avendo un array di lunghezza 5, la chiamata dell'indice 5 o superiore, o -6 o inferiore, non produce risultato.

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 JSON di Redis non supportati

ElastiCache per Redis supporta i seguenti comandi JSON di 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

Accoda uno o più valori rispetto ai valori dell'array nel percorso.

Sintassi

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

- **key** (richiesto) – Una chiave Redis del tipo di documento JSON.
- **path** (richiesto) – Un percorso JSON.
- **json** (richiesto) – Il valore JSON da accodare all'array.

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di numeri interi che rappresentano 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 ricorrenza di un valore JSON scalare negli array nel 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) – Una chiave Redis del tipo di documento JSON.

- **path** (richiesto) – Un percorso JSON.
- **json-scalar** (richiesto) —Il valore scalare per cercare. Lo scalare JSON si riferisce a valori diversi da oggetti o array. Vale a dire che stringa, numero, booleano e null sono valori scalari.
- **start** (facoltativo) – L'indice start, incluso. Se non è fornito, viene utilizzata l'impostazione predefinita, 0.
- **end** (facoltativo) – L'indice end, escluso. Se non è fornito, viene utilizzata l'impostazione predefinita, 0, cioè viene incluso 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

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

Sintassi

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

- **key** (richiesto) – Una chiave Redis del tipo di documento JSON.
- **path** (richiesto) – Un percorso JSON.
- **index** (richiesto) —Un indice array prima del quale sono inseriti valori.
- **json** (richiesto) – Il valore JSON da accodare all'array.

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di numeri interi che rappresentano 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

Ottiene la lunghezza dei valori dell'array nel percorso.

Sintassi

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

- **key** (richiesto) – Una chiave Redis del tipo di documento JSON.
- **path** (facoltativo) – Un percorso JSON. Se non è fornito, per impostazione predefinita viene ripristinata la radice.

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di numeri interi che rappresentano la lunghezza dell'array 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 un elemento nell'indice dall'array. Il prelievo di un array vuoto restituisce null.

Sintassi

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

- **key** (richiesto) – Una chiave Redis del tipo di documento JSON.
- **path** (facoltativo) – Un percorso JSON. Se non è fornito, per impostazione predefinita viene ripristinata la radice.
- **index** (facoltativo) —La posizione nell'array da cui iniziare il prelievo.
 - 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:

- Array di stringhe in blocco che rappresentano i valori prelevati 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 in blocco che rappresenta il valore JSON prelevato.
- 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 un array nel percorso in modo che diventi un array secondario `[start, end]`, 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) – Una chiave Redis del tipo di documento JSON.
- `path` (richiesto) – Un percorso JSON.
- `start` (richiesto) – L'indice `start`, incluso.
- `end` (richiesto) – L'indice `end`, incluso.

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di numeri interi che rappresentano 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

Cancella gli array o un oggetto nel percorso.

Sintassi

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

- **key** (richiesto) – Una chiave Redis del tipo di documento JSON.
- **path** (facoltativo) – Un percorso JSON. Se non è fornito, per impostazione predefinita viene ripristinata la radice.

Valori restituiti

- Numero intero, il numero di container cancellati.
- La cancellazione di un oggetto o un array vuoto rappresenta 1 container cancellato.
- La cancellazione di un valore non container 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) 7
127.0.0.1:6379> JSON.CLEAR k1 $[*]
(integer) 4
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

Riporta informazioni. Sottocomandi supportati:

- **MEMORY** <key> [path] – Riporta l'utilizzo della memoria in byte di un valore JSON. Se non è fornito, viene ripristinato il percorso predefinito, la radice.
- **FIELDS** <key> [path] – 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 messaggi della guida del comando.

Sintassi

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

Dipende dal sottocomando:

MEMORY

- Se il percorso è una sintassi avanzata:
 - Restituisce un array di numeri interi che rappresentano 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, le dimensioni della memoria e il valore JSON in byte.

- Restituisce null se la chiave Redis non esiste.

FIELDS

- Se il percorso è una sintassi avanzata:
 - Restituisce un array di numeri interi che rappresentano il numero di campi 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, il numero di campi del valore JSON.
 - Restituisce null se la chiave Redis non esiste.

HELP – Restituisce un array di messaggi della guida.

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 del documento. Se il percorso è la radice, equivale all'eliminazione della chiave da Redis.

Sintassi

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

- **key (richiesto)** – Una chiave Redis del tipo di documento JSON.

- **path (facoltativo)** – Un percorso JSON. Se non è fornito, per impostazione predefinita viene ripristinata 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

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) – Una chiave Redis del tipo di documento JSON.
- **INDENT/NEWLINE/SPACE** (facoltativo) — Controlla il formato della stringa JSON restituita, cioè , “pretty print”. Il valore predefinito di ciascuno è una stringa vuota. Può essere sovrascritto in qualunque combinazione. Possono essere specificati in qualunque ordine.
- **NOESCAPE** — Facoltativo, presenza consentita per compatibilità legacy senza altri effetti.
- **path** (facoltativo) — Zero o più percorsi JSON; il valore predefinito è la radice, se non viene fornito. 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 con stringify, 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.*
["\n21 2nd Street\n","\nNew York\n","\nNY\n","\n10021-3100\n"]
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" $.address.*
["\n\t\n21 2nd Street\n","\n\t\nNew York\n","\n\t\nNY\n","\n\t\n10021-3100\n\n"]
127.0.0.1:6379> JSON.GET k1 $.firstName $.lastName $.age
{"$.firstName":["\nJohn\n"],"\$.lastName":["\nSmith\n"],"\$.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 $.*
"[{},{\n"a\n":1},{\n"a\n":1,\n"b\n":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
{"\n"street\n":\n"21 2nd Street\n","\n"city\n":\n"New York\n","\n"state\n":\n"NY\n","\n"zipcode\n":
\n"10021-3100\n"}"
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" .address
{"\n\t\n"street\n": \n"21 2nd Street\n","\n\t\n"city\n": \n"New York\n","\n\t\n"state\n": \n"NY\n","\n
\t\n"zipcode\n": \n"10021-3100\n\n"}"
127.0.0.1:6379> JSON.GET k1 .firstName .lastName .age
{"\n".firstName\n":\n"John\n","\n".lastName\n":\n"Smith\n","\n".age\n":27}"
```


JSON.MGET

Ottiene JSON serializzati nel percorso da più chiavi del documento. Restituisce null per una chiave o percorso JSON inesistente.

Sintassi

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

- **key** (richiesto) — Una o più chiavi Redis di tipo documento.
- **path** (richiesto) – Un percorso JSON.

Valori restituiti

- Array di stringhe in blocco. La dimensione dell'array è uguale al numero di chiavi nel comando. Ogni elemento dell'array è popolato con a) JSON serializzato come collocato dal percorso o b) null se la chiave non esiste, il percorso non esiste nel documento oppure 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 nel percorso di un determinato numero.

Sintassi

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

- **key** (richiesto) – Una chiave Redis del tipo di documento JSON.
- **path** (richiesto) – Un percorso JSON.
- **numero** (richiesto) – Un numero.

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di stringhe in blocco che rappresenta 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 in blocco 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\":{\"\"a\":2},\"b\":{\"\"a\":1,\"\"b\":2},\"c\":{\"\"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\":{\"\"a\":2},\"b\":{\"\"a\":2,\"\"b\":3},\"c\":{\"\"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\":{\"\"a\":2},\"b\":{\"\"a\":2,\"\"b\":3},\"c\":{\"\"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 nel percorso di un determinato numero.

Sintassi

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

- **key** (richiesto) – Una chiave Redis del tipo di documento JSON.
- **path** (richiesto) – Un percorso JSON.
- **numero** (richiesto) – Un numero.

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di stringhe in blocco 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 è esterno all'intervallo di numero in virgola mobile a doppia precisione IEEE a 64 bit.
- `NONEXISTENT` se la chiave del documento non esiste.

Se il percorso è una sintassi limitata:

- Stringa in blocco 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 è esterno all'intervallo di un 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

Ottiene il numero di chiavi nei valori oggetto nel percorso.

Sintassi

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

- **key** (richiesto) – Una chiave Redis del tipo di documento JSON.
- **path** (facoltativo) – Un percorso JSON. Se non è fornito, per impostazione predefinita viene ripristinata la radice.

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di numeri interi che rappresentano 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", "c":{"a":3,"b":4}}, "e":1}'
OK
127.0.0.1:6379> JSON.OBJLEN k1 $.a
1) (integer) 0
127.0.0.1:6379> JSON.OBJLEN k1 $.a.*
(empty array)
127.0.0.1:6379> JSON.OBJLEN k1 $.b
1) (integer) 1
127.0.0.1:6379> JSON.OBJLEN k1 $.b.*
1) (nil)
127.0.0.1:6379> JSON.OBJLEN k1 $.c
1) (integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 $.c.*
1) (nil)
2) (nil)
127.0.0.1:6379> JSON.OBJLEN k1 $.d
1) (integer) 3
127.0.0.1:6379> JSON.OBJLEN k1 $.d.*
1) (nil)
2) (nil)
3) (integer) 2
127.0.0.1:6379> JSON.OBJLEN k1 $.*
1) (integer) 0
2) (integer) 1
3) (integer) 2
4) (integer) 3
5) (nil)
```

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

Ottiene i nomi chiave nei valori oggetto nel percorso.

Sintassi

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

- **key (richiesto)** – Una chiave Redis del tipo di documento JSON.
- **path (facoltativo)** – Un percorso JSON. Se non è fornito, per impostazione predefinita viene ripristinata 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 container, la risposta è un array RESP o un array nidificato.

- 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 JSON sono mappate a stringhe in blocco RESP.
- Gli array JSON sono rappresentati come array RESP, dove il primo elemento è la stringa semplice [, seguita dagli elementi dell'array.
- Gli oggetti JSON sono rappresentati come array RESP, dove il primo elemento è la stringa semplice {, seguita da coppie chiave-valore, ognuna delle quali è una stringa in blocco RESP.

Sintassi

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

- **key** (richiesto) – Una chiave Redis del tipo di documento JSON.
- **path** (facoltativo) – Un percorso JSON. Se non è fornito, per impostazione predefinita viene ripristinata 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 RESP 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 valori JSON nel percorso.

Se il percorso richiede un membro oggetto:

- Se l'elemento padre non esiste, il comando restituisce un errore NONEXISTENT.
- Se l'elemento padre esiste ma non è un oggetto, il comando restituisce 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. In caso contrario, il comando restituisce 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 restituisce un errore `NONEXISTENT`.
- Se l'elemento padre esiste ma non è un array, il comando restituisce `ERROR`.
- Se l'elemento padre esiste ma l'indice è esterno ai limiti, il comando restituisce un errore `OUTOFBOUNDARIES`.
- 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 è possibile avere 0 o 1 identificatori [NX | XX].

- `key` (richiesto) – Una chiave Redis del tipo di documento JSON.
- `path` (richiesto) – Un 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. Inserire, cioè, un nuovo documento. Se il percorso non è la radice, impostare il valore solo se il percorso non esiste. Inserire, cioè, un valore nel documento.
- `XX` (facoltativo) – Se il percorso è la radice, impostare il valore solo se la chiave Redis esiste. Sostituire, cioè, il documento esistente. Se il percorso non è la radice, impostare il valore solo se il percorso esiste. Aggiornare, cioè, 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

Accoda un elemento stringa alle stringhe JSON nel percorso.

Sintassi

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

- **key** (richiesto) – Una chiave Redis del tipo di documento JSON.
- **path** (facoltativo) – Un percorso JSON. Se non è fornito, per impostazione predefinita viene ripristinata la radice.
- **json_string** (richiesto) – La rappresentazione JSON di una stringa. Tieni presente che una stringa JSON deve essere tra virgolette. Esempio: "esempio di stringa".

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di numeri interi che rappresentano la nuova lunghezza della stringa in ogni percorso.
- Se un valore nel percorso non è una stringa, il valore restituito corrispondente è null.
- Errore SYNTAXERR se l'argomento 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 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.STRLLEN

Ottiene le lunghezze dei valori stringa JSON nel percorso.

Sintassi

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

- **key** (richiesto) – Una chiave Redis del tipo di documento JSON.
- **path** (facoltativo) – Un percorso JSON. Se non è fornito, per impostazione predefinita viene ripristinata la radice.

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di numeri interi che rappresenta del valore 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

Alterna valori booleani vero e falso nel percorso.

Sintassi

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

- **key** (richiesto) – Una chiave Redis del tipo di documento JSON.
- **path** (facoltativo) – Un percorso JSON. Se non è fornito, per impostazione predefinita viene ripristinata la radice.

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di numeri interi (0 - falso, 1 - vero) che rappresentano valore booleano risultante in ogni percorso.
- Se un valore non è un valore booleano, il valore restituito corrispondente 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.
- Errore WRONGTYPE 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

Reports the type of values in the path indicated.

Sintassi

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

- **key** (richiesto) – Una chiave Redis del tipo di documento JSON.
- **path** (facoltativo) – Un percorso JSON. Se non è fornito, per impostazione predefinita viene ripristinata la radice.

Valori restituiti

Se il percorso è una sintassi avanzata:

- Array di stringhe che rappresentano 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
```

Assegnazione di tag delle risorse ElastiCache

Per semplificare la gestione delle risorse ElastiCache, 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 ElastiCache e monitorare così ogni proprietario dell'istanza 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. Con un set di chiavi di tag coerente, la gestione delle risorse risulta semplificata. Puoi cercare e filtrare le risorse in base ai tag aggiunti. Per ulteriori informazioni sulle modalità di implementazione di una strategia efficace di applicazione di tag alle risorse, consulta il [whitepaper AWS Best practice per l'applicazione di tag](#).

I tag non hanno alcun significato semantico per ElastiCache 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. Inoltre, se si aggiungono o eliminano tag in un gruppo di replica, verranno aggiunti o rimossi anche i tag di tutti i nodi del gruppo di replica.

Puoi lavorare con i tag utilizzando la AWS Management Console, lo AWS CLI e l'API ElastiCache.

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 [Autorizzazioni a livello di risorsa](#).

Risorse cui è possibile associare tag

Puoi assegnare tag alla maggior parte delle risorse ElastiCache 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 al relativo tag Nome ma questo tag non ha un significato semantico per il servizio ElastiCache

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 ElastiCache la CLI AWS o un SDK AWS, puoi usare il parametro Tags sull'operazione API ElastiCache pertinente per applicare i tag. Questi sono:

- `CreateServerlessCache`
- `CreateCacheCluster`
- `CreateReplicationGroup`
- `CopyServerlessCacheSnapshot`
- `CopySnapshot`
- `CreateCacheParameterGroup`
- `CreateCacheSecurityGroup`
- `CreateCacheSubnetGroup`
- `CreateServerlessCacheSnapshot`
- `CreateSnapshot`

- `CreateUserGroup`
- `CreateUser`
- `PurchaseReservedCacheNodesOffering`

Nella seguente tabella sono descritte le risorse ElastiCache a cui puoi associare i tag, nonché le risorse che possono essere associate a tag utilizzando l'API ElastiCache, la CLI AWS o un SDK AWS.

Support dell'assegnazione di tag per le risorse ElastiCache

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

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

Note

Non è possibile assegnare tag ai Global Datastore.

Puoi applicare autorizzazioni basate su tag a livello di risorsa nelle policy IAM alle operazioni dell'API ElastiCache che supportano il l'assegnazione di tag 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 [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](#).

Applicazione di tag a cache e snapshot

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

- CreateReplicationGroup
 - Se il file `--primary-cluster-id` e `--tags` sono inclusi nella richiesta, i tag di richiesta verranno aggiunti al gruppo di replica e propagati a tutti i cluster di cache nel gruppo di replica. Se il cluster di cache primario dispone di tag esistenti, questi verranno sovrascritti con i tag di richiesta per avere tag coerenti in tutti i nodi.

Se non sono presenti tag di richiesta, i tag cluster di cache primaria verranno aggiunti al gruppo di replica e propagati a tutti i cluster di cache.

- Se si specifica `--snapshot-name` o `--serverless-cache-snapshot-name`:

Se i tag sono inclusi nella richiesta, il gruppo di replica viene contrassegnato solo con tali tag. Se nella richiesta non sono inclusi tag, i tag dello snapshot vengono aggiunti al gruppo di replica.

- Se il file `--global-replication-group-id` viene fornito:

Se i tag sono inclusi nella richiesta, i tag della richiesta vengono aggiunti al gruppo di replica e propagati a tutti i cluster di cache.

- **CreateCacheCluster**

- Se il file `--replication-group-id` viene fornito:

Se i tag sono inclusi nella richiesta, il cluster di cache verrà contrassegnato solo con tali tag. Se nella richiesta non sono inclusi tag, il cluster di cache erediterà i tag del gruppo di replica anziché i tag del cluster di cache primario.

- Se il file `--snapshot-name` viene fornito:

Se i tag sono inclusi nella richiesta, il cluster di cache verrà contrassegnato solo con tali tag. Se nella richiesta non sono inclusi tag, i tag snapshot verranno aggiunti al cluster di cache.

- **CreateServerlessCache:**

- Se i tag sono inclusi nella richiesta, solo i tag della richiesta vengono aggiunti alla cache serverless.

- **CreateSnapshot**

- Se il file `--replication-group-id` 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 del gruppo di replica verranno aggiunti allo snapshot.

- Se il file `--cache-cluster-id` 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 di cache verranno aggiunti allo snapshot.

- **Snapshot automatiche**

I tag verranno propagati dai tag del gruppo di replica.

- **CreateServerlessCacheSnapshot:**

- Se i tag sono inclusi nella richiesta, solo i tag della richiesta vengono aggiunti allo snapshot della cache serverless.
- 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.
- CopyServerlessCacheSnapshot:
 - Se i tag sono inclusi nella richiesta, solo i tag della richiesta vengono aggiunti allo snapshot della cache serverless.
- AddTagsToResourceeRemoveTagsFromResource:
 - I tag verranno aggiunti/rimossi dal gruppo di replica e l'operazione verrà propagata a tutti i cluster del gruppo di replica.

Note

AddTagsToResourceeRemoveTagsFromResource non può essere utilizzato per i gruppi di protezione e parametri predefiniti.

- IncreaseReplicaCount e ModifyReplicationGroupShardConfiguration:
 - A tutti i nuovi cluster aggiunti al gruppo di replica vengono applicati gli stessi tag del gruppo di replica.

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 ElastiCache consenta l'uso di qualsiasi carattere nei tag, altri servizi presentano limitazioni più restrittive. 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 ElastiCache che è possibile taggare, vedere [Risorse cui è possibile associare tag](#).

Esempio: assegnazione di tag alle risorse

- Creazione di una cache serverless utilizzando i tag

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine redis \  
  --tags Key="Cost Center", Value="1110001" Key="project",Value="XYZ"
```

- Aggiunta di tag a una cache serverless

```
aws elasticache add-tags-to-resource \  
  --resource-name arn:aws:elasticache:us-east-1:111111222233:serverlesscache:my-cache \  
  --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Aggiunta di tag a un gruppo di replica.

```
aws elasticache add-tags-to-resource \  
  --resource-name arn:aws:elasticache:us-east-1:111111222233:replicationgroup:my-rg \  
  --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Creazione di un cluster di cache con i tag.

```
aws elasticache create-cache-cluster \  
  --cluster-id testing-tags \  
  --cluster-description cluster-test \  
  --cache-subnet-group-name test \  
  --cache-node-type cache.t2.micro \  
  --engine redis \  
  --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```



```
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Creazione di uno snapshot serverless con tag

```
aws elasticache create-serverless-cache-snapshot \
--serverless-cache-name testing-tags \
--serverless-cache-snapshot-name bkp-testing-tags-scs \
--tags Key="work",Value="foo"
```

- Esempio: creazione di uno snapshot con tag

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

```
aws elasticache create-snapshot \
--replication-group-id testing-tags \
--snapshot-name bkp-testing-tags-rg \
--tags Key="work",Value="foo"
```

Esempi di policy di controllo degli accessi basati sui tag

1. Consentire l'operazione AddTagsToResource a un cluster solo se il cluster ha il tag Project=XYZ.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "elasticache:AddTagsToResource",
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "XYZ"
        }
      }
    }
  ]
}
```

- Consentire l'operazione `RemoveTagsFromResource` da un gruppo di replica se contiene i tag progetto e servizio e le chiavi sono diverse da `Project` e `Service`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "elasticache:RemoveTagsFromResource",
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Service": "Elasticache",
          "aws:ResourceTag/Project": "XYZ"
        },
        "ForAnyValue:StringNotEqualsIgnoreCase": {
          "aws:TagKeys": [
            "Project",
            "Service"
          ]
        }
      }
    }
  ]
}
```

- Consentire `AddTagsToResource` a qualsiasi risorsa solo se i tag sono diversi da progetto e servizio.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "elasticache:AddTagsToResource",
      "Resource": [
        "arn:aws:elasticache:*:*:*:*"
      ],
      "Condition": {
        "ForAnyValue:StringNotEqualsIgnoreCase": {
          "aws:TagKeys": [

```

```

        "Service",
        "Project"
      ]
    }
  }
]
}

```

4. Negare l'operazione `CreateReplicationGroup` se la richiesta ha Tag `Project=Foo`.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "elasticache:CreateReplicationGroup",
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/Project": "Foo"
        }
      }
    }
  ]
}

```

5. Negare l'operazione `CopySnapshot` se lo snapshot di fonte ha tag `project=xyz` e il tag `request` è `Service=Elasticache`.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "elasticache:CopySnapshot",
      "Resource": [
        "arn:aws:elasticache:*:*:snapshot:*"
      ],
      "Condition": {
        "StringEquals": {

```

```

        "aws:ResourceTag/Project": "XYZ",
        "aws:RequestTag/Service": "Elasticache"
    }
}
]
}

```

6. Rifiuto dell'azione `CreateCacheCluster` se il tag della richiesta `Project` manca o è diverso da `Dev`, `QA` o `Prod`.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*",
        "arn:aws:elasticache:*:*:securitygroup:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "Null": {
          "aws:RequestTag/Project": "true"
        }
      }
    }
  ],
  {
    "Effect": "Allow",
    "Action": [

```

```
        "elasticache:CreateCacheCluster",
        "elasticache:AddTagsToResource"
    ],
    "Resource": "arn:aws:elasticache:*:*:cluster:*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/Project": [
                "Dev",
                "Prod",
                "QA"
            ]
        }
    }
}
```

Per informazioni correlate sulle chiavi di condizione, consulta [Utilizzo delle chiavi di condizione](#).

Monitoraggio dei costi con i tag di allocazione dei costi

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

Un tag di allocazione dei costi ElastiCache è una coppia chiave-valore definita e associata a una risorsa ElastiCache. 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 `ElastiCache` e `Asia Pacific (Singapore)`, mentre su un'altra risorsa potresti avere i valori `ElastiCache` e `Europe (Frankfurt)`. Puoi quindi vedere i tuoi costi totali ElastiCache 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 ElastiCache ai nodi Redis. Quando aggiungi, elenchi, modifichi, copi o rimuovi un tag, l'operazione viene applicata solo al nodo specificato.

Caratteristiche dei tag di allocazione dei costi di ElastiCache

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

ARN di esempio: `arn:aws:elasticache:<region>:<customer-id>:<resource-type>:<resource-name>`

Arn di esempio: `arn:aws:elasticache:us-west-2:1234567890: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 ElastiCache 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 `ElastiCache`.

AWS non applica significati semantici ai tag. I tag vengono interpretati rigorosamente come stringhe di caratteri. AWS non imposta automaticamente i tag sulle risorse ElastiCache.

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:elasticache:us-west-2:1234567890:cluster:my-cluster`

I costi di allocazione dei tag vengono applicati ai nodi ElastiCache for Redis. Il nodo a cui aggiungere tag viene specificato mediante un Amazon Resource Name (ARN).

Arn di esempio `arn:aws:elasticache:us-west-2:1234567890: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

È possibile utilizzare AWS CLI per elencare i tag su una risorsa esistente ElastiCache tramite l'operazione [list-tags-for-resource](#).

Il codice seguente utilizza l'AWS CLI per elencare i tag sul nodo Redis `my-cluster-001` nel cluster `my-cluster` nella regione `us-west-2`.

Per Linux, macOS o Unix:

```
aws elasticache list-tags-for-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

Per Windows:

```
aws elasticache list-tags-for-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

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 nella risposta, l'operazione restituisce una mappa TagList vuota.

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

Per ulteriori informazioni sulle proprietà di ElastiCache [list-tags-for-resource](#), consultare AWS CLI.

Aggiunta di tag utilizzando la AWS CLI

È possibile utilizzare la AWS CLI per aggiungere i tag a una risorsa ElastiCache esistente tramite l'operazione CLI [add-tags-to-resource](#). 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 la AWS CLI per aggiungere le chiavi Service e Region con i valori elasticache e us-west-2 rispettivamente sul nodo my-cluster-001 nel cluster my-cluster nella regione us-west-2.

Per Linux, macOS o Unix:

```
aws elasticache add-tags-to-resource \
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 \
  --tags Key=Service,Value=elasticache \
  Key=Region,Value=us-west-2
```


Per Windows:

```
aws elasticache add-tags-to-resource ^
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 ^
  --tags Key=Service,Value=elasticache ^
         Key=Region,Value=us-west-2
```

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

```
{
  "TagList": [
    {
      "Value": "elasticache",
      "Key": "Service"
    },
    {
      "Value": "us-west-2",
      "Key": "Region"
    }
  ]
}
```

Per ulteriori informazioni sulle proprietà di ElastiCache [add-tags-to-resource](#), consultare AWS CLI.

Puoi anche utilizzare l'AWS CLI per aggiungere i tag a un cluster quando crei un nuovo cluster mediante l'operazione [create-cache-cluster](#). Non è possibile aggiungere tag durante la creazione di un cluster mediante la console di gestione ElastiCache. Dopo aver creato il cluster, puoi quindi utilizzare la console per aggiungere i tag al cluster.

Modifica dei tag utilizzando l' AWS CLI

Puoi utilizzare la AWS CLI per modificare i tag su un nodo in un cluster ElastiCache.

Per modificare i tag:

- Utilizzare [add-tags-to-resource](#) per aggiungere un nuovo tag e valore o per modificare il valore associato a un tag esistente.
- Utilizzare [remove-tags-from-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

È possibile utilizzare la AWS CLI per rimuovere i tag da un nodo esistente in un cluster ElastiCache per Redis esistente tramite l'operazione [remove-tags-from-resource](#).

Il codice seguente utilizza la AWS CLI per rimuovere i tag con le chiavi Service e Region dal nodo `my-cluster-001` nel cluster `my-cluster` nella regione `us-west-2`.

Per Linux, macOS o Unix:

```
aws elasticache remove-tags-from-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 \  
  --tag-keys PM Service
```

Per Windows:

```
aws elasticache remove-tags-from-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 ^  
  --tag-keys PM 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 sulle proprietà di ElastiCache [remove-tags-from-resource](#), consultare AWS CLI.

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

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

I tag di allocazione dei costi vengono applicati ai cluster ElastiCache for Memcached. Il cluster a cui aggiungere tag viene specificato mediante un Amazon Resource Name (ARN).

Arn di esempio `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Argomenti

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

Elenco dei tag con l'API ElastiCache

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

Il codice seguente utilizza l'API ElastiCache per elencare i tag sulla risorsa nella regione us-west-2 `my-cluster-001`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ListTagsForResource  
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Version=2015-02-02  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Aggiunta di tag con l'API ElastiCache

È possibile utilizzare l'API ElastiCache per aggiungere i tag a un cluster ElastiCache esistente tramite l'operazione [AddTagsToResource](#). 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 ElastiCache per aggiungere le chiavi Service e Region con i valori `elasticache` e `us-west-2` rispettivamente per le risorse `my-cluster-001` nella regione `us-west-2`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=AddTagsToResource  
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256
```

```
&Tags.member.1.Key=Service
&Tags.member.1.Value=elasticache
&Tags.member.2.Key=Region
&Tags.member.2.Value=us-west-2
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Per ulteriori informazioni, consulta [AddTagsToResource](#) nella Guida di riferimento delle API di Amazon ElastiCache.

Modifica dei tag utilizzando l'API ElastiCache

Puoi utilizzare l'API ElastiCache per modificare i tag su un cluster ElastiCache.

Per modificare il valore di un tag:

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

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

Utilizzare [RemoveTagsFromResource](#) per rimuovere i tag dalla risorsa.

Rimozione dei tag utilizzando l'API ElastiCache

È possibile utilizzare l'API ElastiCache per rimuovere i tag da un nodo ElastiCache for Redis esistente tramite l'operazione [RemoveTagsFromResource](#).

Il codice seguente utilizza l'API ElastiCache per rimuovere i tag con le chiavi Service e Region dal `nodomy-cluster-001` nel `clustermymy-cluster` nella regione `us-west-2`.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=RemoveTagsFromResource
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&TagKeys.member.1=Service
&TagKeys.member.2=Region
```

```
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Utilizzo di Amazon ElastiCache Well-Architected Lens

Questa sezione descrive Amazon ElastiCache Well-Architected Lens, una raccolta di principi e linee guida per la progettazione di carichi di lavoro ElastiCache correttamente strutturati.

- ElastiCache Lens è in aggiunta a [Framework AWS Well-Architected](#).
- Ogni pilastro contiene una serie di domande per aiutare ad avviare la discussione su un'architettura ElastiCache.
- Ogni domanda contiene una serie di procedure principali insieme ai relativi punteggi per la segnalazione.
 - **Obbligatorio:** la procedura obbligatoria prima di passare in produzione (se non viene eseguita comporta un alto rischio)
 - **Best practice:** la migliore procedura per un cliente
 - **Consigliato:** la procedura consigliata ai clienti (se non viene eseguita comporta un rischio medio)
- Terminologia di Well-Architected
 - **Componente:** codice, configurazione e risorse AWS che insieme soddisfano un requisito. I componenti interagiscono con altri componenti e spesso equivalgono a un servizio nelle architetture di microservizi.
 - **Carico di lavoro:** un insieme di componenti che forniscono valore aziendale. Esempi di carichi di lavoro sono siti Web di marketing, siti Web di e-commerce, il back-end di un'app per dispositivi mobili, piattaforme di analisi e così via.

Argomenti

- [Pilastro dell'eccellenza operativa di Amazon ElastiCache Well-Architected Lens](#)
- [Pilastro della sicurezza di Amazon ElastiCache Well-Architected Lens](#)
- [Pilastro dell'affidabilità di Amazon ElastiCache Well-Architected Lens](#)
- [Pilastro dell'efficienza delle prestazioni di Amazon ElastiCache Well-Architected Lens](#)
- [Pilastro dell'ottimizzazione dei costi di Amazon ElastiCache Well-Architected Lens](#)

Pilastro dell'eccellenza operativa di Amazon ElastiCache Well-Architected Lens

Il pilastro dell'eccellenza operativa si concentra sull'esecuzione e sul monitoraggio dei sistemi per fornire valore aziendale e migliorare continuamente processi e procedure. Gli argomenti chiave includono l'automazione delle modifiche, la risposta agli eventi e la definizione degli standard per gestire le operazioni quotidiane.

Argomenti

- [EO 1: come si fa a comprendere e rispondere agli avvisi e agli eventi generati dal cluster ElastiCache?](#)
- [EO 2: quando e come si dimensionano i cluster ElastiCache esistenti?](#)
- [EO 3: come si gestiscono le risorse del cluster ElastiCache e si mantiene aggiornato il cluster?](#)
- [EO 4: come si gestiscono le connessioni dei client ai cluster ElastiCache?](#)
- [EO 5: come si distribuiscono i componenti ElastiCache per un carico di lavoro?](#)
- [EO 6: come si pianificano e si contengono gli errori?](#)
- [EO 7: come si risolvono gli eventi del motore Redis?](#)

EO 1: come si fa a comprendere e rispondere agli avvisi e agli eventi generati dal cluster ElastiCache?

Introduzione della domanda: quando si utilizzano i cluster ElastiCache, è possibile ricevere notifiche e avvisi opzionali quando si verificano eventi specifici. ElastiCache, per impostazione predefinita, registra [gli eventi](#) relativi alle risorse, come il failover, la sostituzione dei nodi, le operazioni di dimensionamento, la manutenzione programmata e altro ancora. Ogni evento include la data e l'ora, il nome e il tipo di origine e una descrizione.

Vantaggio della domanda: la capacità di comprendere e gestire i motivi alla base degli eventi che generano gli avvisi del cluster consente di operare in modo più efficace e di rispondere agli eventi in modo appropriato.

- [Obbligatorio] Rivedi gli eventi generati da ElastiCache sulla console ElastiCache (dopo aver selezionato la tua regione) o utilizzando il comando [describe-events](#) dell'[Interfaccia della linea di comando Amazon](#) (CLI AWS) e l'[API di ElastiCache](#). Configura ElastiCache in modo tale che invii notifiche sugli eventi importanti del cluster con Amazon Simple Notification Service (Amazon SNS).

L'utilizzo di Amazon SNS con i cluster ti consente di intraprendere azioni programmatiche sugli eventi ElastiCache.

- Esistono due grandi categorie di eventi: eventi attuali e programmati. L'elenco degli eventi correnti include: creazione ed eliminazione delle risorse, operazioni di dimensionamento, failover, riavvio del nodo, creazione di snapshot, modifica dei parametri del cluster, rinnovo del certificato CA, eventi di errore (errore di provisioning del cluster per VPC o ENI, errori di dimensionamento per ENI ed errori di snapshot). L'elenco degli eventi pianificati include: nodo programmato per la sostituzione durante la finestra di manutenzione e sostituzione del nodo riprogrammata.
- Sebbene non sia necessario reagire immediatamente ad alcuni di questi eventi, è fondamentale esaminare tutti gli eventi di errore:
 - ElastiCache:AddCacheNodeFailed
 - ElastiCache:CacheClusterProvisioningFailed
 - Elasticache: cacheclusterScalingFailed
 - ElastiCache:cachenodesRebooted
 - ElastiCache: SnapshotFailed (solo Redis)
- [Risorse]:
 - [Gestione ElastiCache Amazon SNS](#)
 - [Notifiche di eventi Amazon SNS](#)
- [Best practice] Per automatizzare le risposte agli eventi, utilizza le funzionalità di prodotti e servizi AWS come SNS e funzioni Lambda. Segui le best practice apportando modifiche piccole, frequenti e reversibili, come codice per migliorare le tue operazioni nel tempo. Utilizza le metriche di Amazon CloudWatch per monitorare i cluster.

[Risorse]: [monitora gli endpoint di replica di lettura di Amazon ElastiCache per Redis \(modalità cluster disabilitata\) con AWS Lambda, Amazon Route 53 e Amazon SNS](#) per un caso d'uso che utilizza Lambda e SNS.

EO 2: quando e come si dimensionano i cluster ElastiCache esistenti?

Introduzione della domanda: il corretto dimensionamento del cluster ElastiCache è un atto di bilanciamento che deve essere valutato ogni volta che vengono apportate modifiche ai tipi di carico di lavoro sottostanti. Il tuo obiettivo è operare con l'ambiente delle dimensioni giuste per il tuo carico di lavoro.

Vantaggio della domanda: l'eccessivo utilizzo delle risorse può comportare una latenza elevata e una riduzione complessiva delle prestazioni. Il sottoutilizzo, invece, può comportare un sovradimensionamento delle risorse a fronte di un'ottimizzazione dei costi non ottimale. Dimensionando correttamente gli ambienti, è possibile trovare un equilibrio tra efficienza delle prestazioni e ottimizzazione dei costi. Per ovviare all'utilizzo eccessivo o insufficiente delle risorse, ElastiCache è dimensionabile in due dimensioni. È possibile dimensionare verticalmente aumentando o diminuendo la capacità del nodo. Puoi anche dimensionare orizzontalmente aggiungendo e rimuovendo nodi.

- [Obbligatorio] L'eccessivo utilizzo della CPU e della rete sui nodi primari deve essere risolto determinando e reindirizzando le operazioni di lettura ai nodi di replica. Usa i nodi di replica per le operazioni di lettura per ridurre l'utilizzo dei nodi primari. Puoi configurarli nella libreria client Redis connettendoti all'endpoint di lettura ElastiCache per la modalità cluster disabilitata o utilizzando il comando READONLY di Redis per la modalità cluster abilitata.

[Risorse]:

- [Individuazione degli endpoint di connessione](#)
- [Cluster Right-Sizing](#)
- [READONLY](#)
- [Obbligatorio] Monitora l'utilizzo delle risorse critiche del cluster come CPU, memoria e rete. L'utilizzo di queste risorse specifiche del cluster deve essere monitorato per determinare la decisione e il tipo di operazione di dimensionamento. Se la modalità cluster ElastiCache per Redis è disabilitata, i nodi primari e di replica possono essere dimensionati verticalmente. I nodi di replica possono anche dimensionare orizzontalmente da 0 a 5 nodi. Per la modalità cluster abilitata, lo stesso vale per ogni partizione del cluster. Inoltre, puoi aumentare o ridurre il numero di partizioni.

[Risorse]:

- [Monitoring best practices with Amazon ElastiCache for Redis using Amazon CloudWatch](#)
- [Dimensionamento ElastiCache per i cluster Redis](#)
- [Dimensionamento ElastiCache per i cluster Memcached](#)
- [Best practice] Il monitoraggio delle tendenze nel tempo può aiutarti a rilevare i cambiamenti del carico di lavoro che rimarrebbero inosservati se monitorati solo in un determinato momento. Per rilevare tendenze a lungo termine, utilizza le metriche di CloudWatch per analizzare intervalli di tempo più lunghi. I risultati derivanti dall'osservazione di periodi prolungati delle metriche di CloudWatch influenzano le previsioni sull'utilizzo delle risorse del cluster. I punti dati e le metriche di CloudWatch sono disponibili per un massimo di 455 giorni.

[Risorse]:

- [Monitoraggio dell'utilizzo con i parametri di CloudWatch](#)
 - [Monitoring Memcached with CloudWatch Metrics](#)
 - [Monitoring best practices with Amazon ElastiCache for Redis using Amazon CloudWatch](#)
- [Best practice] Se le risorse ElastiCache sono create con CloudFormation, è consigliabile eseguire modifiche utilizzando i modelli CloudFormation per preservare la coerenza operativa ed evitare modifiche alla configurazione non gestite e spostamenti dello stack.

[Risorse]:

- [ElastiCache resource type reference for CloudFormation](#)
- [Best practice] Automatizza le operazioni di dimensionamento utilizzando i dati operativi del cluster e definisci soglie in CloudWatch per configurare gli allarmi. Usa Eventi CloudWatch e Simple Notification Service (SNS) per attivare le funzioni Lambda ed eseguire un'API ElastiCache per dimensionare automaticamente i cluster. Un esempio potrebbe essere aggiungere una partizione al cluster quando la metrica EngineCPUUtilization raggiunge l'80% per un lungo periodo di tempo. Un'altra opzione potrebbe essere utilizzare DatabaseMemoryUsedPercentages per una soglia basata sulla memoria.

[Risorse]:

- [Using Amazon CloudWatch Alarms](#)
- [Che cos'è Amazon CloudWatch Events?](#)
- [Utilizzo di AWS Lambda con Amazon SNS](#)
- [Informazioni di riferimento ElastiCache](#)

EO 3: come si gestiscono le risorse del cluster ElastiCache e si mantiene aggiornato il cluster?

Introduzione della domanda: quando si opera su larga scala, è essenziale essere in grado di individuare e identificare tutte le risorse ElastiCache. Quando si implementano nuove funzionalità dell'applicazione, è necessario creare una simmetria delle versioni del cluster in tutti i tipi di ambiente ElastiCache: sviluppo, test e produzione. Gli attributi delle risorse consentono di separare gli ambienti per diversi obiettivi operativi, ad esempio quando si implementano nuove funzionalità e si abilitano nuovi meccanismi di sicurezza.

Vantaggio della domanda: la separazione degli ambienti di sviluppo, test e produzione è una best practice operativa. È inoltre consigliabile che ai cluster e ai nodi in tutti gli ambienti vengano applicate le patch software più recenti utilizzando i processi appresi e documentati. Sfruttare le funzionalità native di ElastiCache consente al team di progettazione di concentrarsi sul raggiungimento degli obiettivi aziendali e non sulla manutenzione di ElastiCache.

- [Best practice] Esegui l'ultima versione del motore disponibile e applica gli aggiornamenti self-service non appena vengono rilasciati. ElastiCache aggiorna automaticamente l'infrastruttura sottostante durante la finestra di manutenzione specificata del cluster. Tuttavia, i nodi in esecuzione nei cluster vengono aggiornati tramite aggiornamenti self-service. Questi aggiornamenti possono essere di due tipi: patch di sicurezza o aggiornamenti software secondari. Assicurati di comprendere la differenza tra i tipi di patch e quando vengono applicate.

[Risorse]:

- [Aggiornamenti di servizio in ElastiCache for Redis](#)
- [Pagina di assistenza agli aggiornamenti del servizio e alla manutenzione gestita da Amazon ElastiCache](#)
- [Best practice] Organizza le risorse ElastiCache utilizzando i tag. Usa i tag sui gruppi di replica e non sui singoli nodi. È possibile configurare i tag in modo che vengano visualizzati quando si eseguono query sulle risorse e utilizzare i tag per eseguire ricerche e applicare filtri. È consigliabile utilizzare i gruppi di risorse per creare e gestire facilmente le raccolte di risorse che condividono set di tag comuni.

[Risorse]:

- [Tagging Best Practices](#)
- [ElastiCache resource type reference for CloudFormation](#)
- [Gruppi di parametri](#)

EO 4: come si gestiscono le connessioni dei client ai cluster ElastiCache?

Introduzione della domanda: quando si opera su larga scala, è necessario capire in che modo i client si connettono al cluster ElastiCache per gestire gli aspetti operativi delle applicazioni (come i tempi di risposta).

Vantaggio della domanda: la scelta del meccanismo di connessione più appropriato garantisce che l'applicazione non si disconnetta a causa di errori di connettività, come i timeout.

- [Obbligatorio] Separa le operazioni di lettura da quelle di scrittura e connettiti ai nodi di replica per eseguire le operazioni di lettura. Tuttavia, tieni presente che quando separi le scritture dalle letture, perdi la capacità di leggere una chiave subito dopo averla scritta a causa della natura asincrona della replica Redis. Il comando WAIT può essere utilizzato per migliorare la sicurezza dei dati nel mondo reale e forzare le repliche a riconoscere le scritture prima di rispondere ai client, a un costo di prestazioni complessivo. L'utilizzo dei nodi di replica per le operazioni di lettura può essere configurato nella libreria client ElastiCache per Redis utilizzando l'endpoint di lettura ElastiCache per la modalità cluster disabilitata. Per la modalità cluster abilitata, utilizza il comando READONLY di ElastiCache per Redis. Per molte delle librerie client ElastiCache per Redis, il comando READONLY di ElastiCache per Redis è implementato per impostazione predefinita o tramite un'impostazione di configurazione.

[Risorse]:

- [Individuazione degli endpoint di connessione](#)
- [READONLY](#)
- [Obbligatorio] Usa il pool di connessioni. La creazione di una connessione TCP comporta un costo in termini di tempo di CPU sia sul lato client che su quello server e il pool consente di riutilizzare la connessione TCP.

Per ridurre il sovraccarico della connessione, è necessario utilizzare il pool di connessioni. Con un pool di connessioni, l'applicazione può riutilizzare e rilasciare connessioni "secondo le necessità", senza il costo di stabilire la connessione. Puoi implementare il pool di connessioni tramite la libreria client ElastiCache per Redis (se supportata), con un framework disponibile per il tuo ambiente applicativo, oppure crearlo da zero.

- [Best practice] Assicurati che il timeout del socket del client sia impostato su almeno un secondo (rispetto al tipico valore predefinito "nessuno" in diversi client).
 - L'impostazione di un valore troppo basso può causare possibili timeout quando il carico del server è elevato. Se si imposta un valore troppo alto, l'applicazione può impiegare molto tempo per rilevare i problemi di connessione.
 - Controlla il volume delle nuove connessioni implementando il pool di connessioni nell'applicazione client. Ciò riduce la latenza e l'utilizzo della CPU necessari per aprire e chiudere le connessioni ed eseguire un handshake TLS se TLS è abilitato sul cluster.

[Risorse]: [Configure Amazon ElastiCache for Redis for higher availability](#)

- [Positivo] L'utilizzo delle pipeline (quando i casi d'uso lo consentono) può aumentare significativamente le prestazioni.

- Con le pipeline si riduce il Round-Trip Time (RTT) tra i client delle applicazioni e il cluster e le nuove richieste possono essere elaborate anche se il client non ha ancora letto le risposte precedenti.
- Con le pipeline puoi inviare più comandi al server senza attendere le risposte e le conferme. L'aspetto negativo delle pipeline è che quando alla fine recuperi tutte le risposte in blocco, potrebbe essere restituito un errore che non è riscontrabile fino alla fine.
- Implementa i metodi per riprovare le richieste quando viene restituito un errore che omette la richiesta non valida.

[Risorse]: [Redis pipelining](#)

EO 5: come si distribuiscono i componenti ElastiCache per un carico di lavoro?

Introduzione della domanda: gli ambienti ElastiCache possono essere distribuiti manualmente tramite la console AWS o in modo programmatico tramite API, CLI, toolkit e così via. Le best practice dell'eccellenza operativa suggeriscono di automatizzare le implementazioni tramite il codice ogni volta che è possibile. Inoltre, i cluster ElastiCache possono essere isolati in base al carico di lavoro o combinati per scopi di ottimizzazione dei costi.

Vantaggio della domanda: la scelta del meccanismo di implementazione più appropriato per gli ambienti ElastiCache può migliorare l'eccellenza operativa nel tempo. Ti consigliamo di eseguire operazioni sotto forma di codice ogni volta che è possibile per ridurre al minimo l'errore umano e aumentare la ripetibilità, la flessibilità e i tempi di risposta agli eventi.

Comprendendo i requisiti di isolamento del carico di lavoro, puoi scegliere di avere ambienti ElastiCache dedicati per carico di lavoro o unire più carichi di lavoro in singoli cluster o in una combinazioni di questi. Comprendere i compromessi può aiutare a trovare un equilibrio tra eccellenza operativa e ottimizzazione dei costi.

- [Obbligatorio] Comprendi le opzioni di implementazione disponibili per ElastiCache e automatizza queste procedure ogni volta che è possibile. I possibili modi di automazione includono CloudFormation, CLI/SDK AWS e API.

[Risorse]:

- [Amazon ElastiCache resource type reference](#)
- [elasticache](#)
- [Informazioni sull'API di Amazon ElastiCache](#)

- [Obbligatorio] Per tutti i carichi di lavoro, determina il livello di isolamento del cluster necessario.
 - [Best practice]: isolamento elevato, una mappatura 1:1 del carico di lavoro ai cluster. Consente un controllo granulare su accesso, dimensioni, dimensionamento e gestione delle risorse ElastiCache in base al carico di lavoro.
 - [Consigliato]: isolamento medio, M:1 isolato per scopo ma forse condiviso tra più carichi di lavoro (ad esempio un cluster dedicato alla memorizzazione nella cache dei carichi di lavoro e un altro dedicato alla messaggistica).
 - [Positivo]: isolamento basso, M:1 tutti gli scopi e completamente condiviso. Consigliato per carichi di lavoro in cui è accettabile l'accesso condiviso.

EO 6: come si pianificano e si contengono gli errori?

Introduzione della domanda: l'eccellenza operativa include l'anticipazione degli errori eseguendo regolarmente esercitazioni "pre mortem" per identificare le potenziali fonti di errore in modo che possano essere rimosse o contenute. ElastiCache offre un'API di failover che consente di simulare eventi di errore dei nodi, a scopo di test.

Vantaggio della domanda: testando in anticipo gli scenari di errore, puoi scoprire in che modo influiscono sul tuo carico di lavoro. Ciò ti consente di testare in sicurezza le procedure di risposta e la loro efficacia, oltre a familiarizzare con l'esecuzione.

[Obbligatorio] Esegui regolarmente test di failover negli account di sviluppo/test. [TestFailover](#)

EO 7: come si risolvono gli eventi del motore Redis?

Introduzione della domanda: l'eccellenza operativa richiede la capacità di analizzare le informazioni a livello di servizio e a livello di motore per capire l'integrità e lo stato dei cluster. Amazon ElastiCache per Redis può inviare i log del motore Redis sia ad Amazon CloudWatch che ad Amazon Kinesis Data Firehose.

Vantaggio della domanda: l'attivazione dei log dei motori Redis sui cluster Amazon ElastiCache per Redis fornisce informazioni sugli eventi che influiscono sullo stato e sulle prestazioni dei cluster. I log del motore Redis forniscono dati direttamente dal motore Redis che non sono disponibili tramite il meccanismo degli eventi ElastiCache. Attraverso un'attenta osservazione degli eventi ElastiCache (vedi il passaggio precedente EO 1) e dei log del motore Redis, è possibile determinare un ordine di eventi per la risoluzione dei problemi sia dal punto di vista del servizio ElastiCache che del motore Redis.

- [Obbligatorio] Assicurati che sia abilitata la funzionalità di registrazione del motore Redis, disponibile in ElastiCache per Redis 6.2 e versioni successive. Questa operazione può essere eseguita durante la creazione del cluster o modificando il cluster dopo la creazione.
- Determina se File di log Amazon CloudWatch o Amazon Kinesis Data Firehose sono la destinazione appropriata per i log dei motori Redis.
- Seleziona un log di destinazione appropriato in CloudWatch o Kinesis Data Firehose per salvare in modo persistente i log. Se disponi di più cluster, considera un log di destinazione diverso per ogni cluster, in quanto ciò contribuisce a isolare i dati per la risoluzione dei problemi.

[Risorse]:

- Consegna dei registri: [Consegna dei registri](#)
- Destinazioni dei log: [File di log Amazon CloudWatch](#)
- Introduzione a File di log Amazon CloudWatch: [Che cos'è Amazon CloudWatch?](#)
- Introduzione ad Amazon Kinesis Data Firehose: [What Is Amazon Kinesis Data Firehose?](#)
- [Best practice] Se usi File di log Amazon CloudWatch, valuta la possibilità di utilizzare Logs Insights di Amazon CloudWatch per eseguire le query sul log del motore Redis per ottenere informazioni importanti.

Ad esempio, crea una query per il gruppo di log CloudWatch contenente i log del motore Redis che restituiscono eventi con un LogLevel pari a "WARNING", ad esempio:

```
fields @timestamp, LogLevel, Message
| sort @timestamp desc
| filter LogLevel = "WARNING"
```

[Risorse]: [Analisi dei dati di log con Amazon CloudWatch Logs Insights](#)

Pilastro della sicurezza di Amazon ElastiCache Well-Architected Lens

Il pilastro della sicurezza si concentra sulla protezione delle informazioni e dei sistemi. Gli argomenti chiave includono la riservatezza e l'integrità dei dati, l'identificazione e la gestione di chi può fare cosa mediante la gestione basata su privilegi, la protezione dei sistemi e l'istituzione di controlli per rilevare gli eventi di sicurezza.

Argomenti

- [SIC 1: quali sono le misure in uso per controllare l'accesso autorizzato ai dati di ElastiCache?](#)

- [SIC 2: le applicazioni richiedono per ElastiCache un'autorizzazione aggiuntiva oltre ai controlli basati sulla rete?](#)
- [SIC 3: esiste il rischio che i comandi possano essere eseguiti inavvertitamente, causando la perdita di dati o errori nei dati?](#)
- [SIC 4: come garantire la crittografia dei dati a riposo con ElastiCache](#)
- [SIC 5: come si crittografano i dati in transito con ElastiCache?](#)
- [SIC 6: come si limita l'accesso alle risorse del piano di controllo \(control-plane\)?](#)
- [SIC 7: come si rileva e si risponde agli eventi di sicurezza?](#)

SIC 1: quali sono le misure in uso per controllare l'accesso autorizzato ai dati di ElastiCache?

Introduzione della domanda: tutti i cluster ElastiCache sono progettati per essere accessibili da istanze Amazon Elastic Compute Cloud in un VPC, funzioni serverless (AWS Lambda) o container (Amazon Elastic Container Service). Lo scenario più comune è accedere a un cluster ElastiCache da un'istanza Amazon Elastic Compute Cloud nello stesso Amazon Virtual Private Cloud (Amazon VPC). Prima di poter eseguire la connessione a un cluster da un'istanza Amazon EC2, devi autorizzare l'istanza Amazon EC2 ad accedere al cluster. Per accedere a un cluster ElastiCache in esecuzione in un VPC, è necessario fornire l'ingresso della rete nel cluster.

Vantaggio della domanda: l'ingresso della rete nel cluster è controllato tramite i gruppi di sicurezza del VPC. Un gruppo di sicurezza funge da firewall virtuale per le istanze Amazon EC2 per controllare il traffico in entrata e quello in uscita. Le regole in entrata controllano il traffico in entrata verso l'istanza e le regole in uscita controllano il traffico in uscita dall'istanza. Nel caso di ElastiCache, quando si avvia un cluster, è necessario associare un gruppo di sicurezza. In tal modo si garantisce che le regole del traffico in entrata e in uscita siano in atto per tutti i nodi che costituiscono il cluster. Inoltre, ElastiCache è configurato per implementare esclusivamente su sottoreti private, in modo da essere accessibile solo tramite la rete privata del VPC.

- [Obbligatorio] Il gruppo di sicurezza associato al cluster controlla l'ingresso e l'accesso della rete al cluster. Per impostazione predefinita, un gruppo di sicurezza non ha alcuna regola in entrata definita e, pertanto, nessun percorso di ingresso a ElastiCache. Per abilitarlo, configura una regola in entrata per il gruppo di sicurezza che specifichi l'indirizzo/intervallo IP di origine, il traffico di tipo TCP e la porta del cluster ElastiCache (ad esempio, la porta predefinita 6379 per ElastiCache per Redis). Sebbene sia possibile consentire un set molto ampio di origini di ingresso, come tutte le risorse all'interno di un VPC (0.0.0.0/0), si consiglia di essere il più granulari possibile nella

definizione delle regole in entrata, ad esempio autorizzare solo l'accesso in entrata ai client Redis in esecuzione su istanze Amazon EC2 associate a uno specifico gruppo di sicurezza.

[Risorse]:

- [Sottoreti e gruppi di sottoreti](#)
- [Accesso al cluster o gruppo di replica](#)
- [Controlla il traffico verso le risorse utilizzando gruppi di sicurezza](#)
- [Gruppi di sicurezza Amazon EC2 per istanze Linux](#)
- [Obbligatorio] Le policy AWS Identity and Access Management possono essere assegnate a funzioni AWS Lambda che possono accedere ai dati ElastiCache. Per abilitare questa funzionalità, crea un ruolo di esecuzione IAM con l'autorizzazione `AWSLambdaVPCLambdaAccessExecutionRole`, quindi assegna il ruolo alla funzione AWS Lambda.

[Risorse]: Configurazione di una funzione Lambda per l'accesso a Amazon ElastiCache in un Amazon VPC: [Tutorial: Configurazione di una funzione Lambda per l'accesso a Amazon ElastiCache in un Amazon VPC](#)

SIC 2: le applicazioni richiedono per ElastiCache un'autorizzazione aggiuntiva oltre ai controlli basati sulla rete?

Introduzione della domanda: negli scenari in cui è necessario limitare o controllare l'accesso ai cluster ElastiCache per Redis a livello di singolo client, si consiglia di eseguire l'autenticazione tramite il comando AUTH di ElastiCache per Redis. I token di autenticazione ElastiCache per Redis, con gestione opzionale di utenti e gruppi di utenti, consentono a ElastiCache per Redis di richiedere una password prima di consentire ai client di eseguire comandi e chiavi di accesso, migliorando così la sicurezza del piano dati.

Vantaggio della domanda: per una migliore sicurezza dei dati, ElastiCache per Redis fornisce meccanismi di protezione dall'accesso non autorizzato ai dati, tra cui l'esecuzione del comando AUTH del controllo degli accessi basato su ruoli (RBAC) o del token AUTH (password) da parte dei client per connettersi a ElastiCache prima di eseguire comandi autorizzati.

- [Best practice] Per ElastiCache per Redis 6.x e versioni successive, determina i controlli di autenticazione e autorizzazione definendo gruppi di utenti, utenti e stringhe di accesso. Assegna gli utenti ai gruppi di utenti, quindi assegna i gruppi di utenti ai cluster. Per utilizzare il RBAC, è necessario selezionarlo al momento della creazione del cluster e abilitare la crittografia in transito. Assicurati di usare un client Redis che supporti TLS per poter utilizzare RBAC.

[Risorse]:

- [Applicazione di RBAC a un gruppo di replica per ElastiCache for Redis](#)
- [Specifica delle autorizzazioni mediante una stringa di accesso](#)
- [ACL](#)
- [ElastiCache supportate per le versioni Redis](#)
- [Best practice] Per le versioni di ElastiCache per Redis precedenti alla 6.x, oltre a impostare una password/token complessa e mantenere una policy rigorosa per le password per il comando AUTH di ElastiCache per Redis, è consigliabile ruotare la password/token. ElastiCache può gestire fino a due (2) token di autenticazione in qualsiasi momento. Puoi anche modificare il cluster per richiedere esplicitamente l'uso di token di autenticazione.

[Risorse]: [Modifica del token AUTH su un cluster ElastiCache per Redis esistente](#)

SIC 3: esiste il rischio che i comandi possano essere eseguiti inavvertitamente, causando la perdita di dati o errori nei dati?

Introduzione della domanda: esistono numerosi comandi Redis che possono avere impatti negativi sulle operazioni se eseguiti per errore o da malintenzionati. Questi comandi possono avere conseguenze impreviste dal punto di vista delle prestazioni e della sicurezza dei dati. Ad esempio, uno sviluppatore che esegue regolarmente il comando FLUSHALL in un ambiente di sviluppo potrebbe per errore chiamare inavvertitamente questo comando in un sistema di produzione, con conseguente perdita accidentale di dati.

Vantaggio della domanda: a partire da ElastiCache per Redis 5.0.3, è possibile su ElastiCache rinominare determinati comandi che potrebbero disturbare il carico di lavoro. La ridenominazione dei comandi può aiutare a evitare che vengano eseguiti inavvertitamente sul cluster.

- [Obbligatorio]

[Risorse]:

- [ElastiCache per Redis versione 5.0.3 \(obsoleta, usa versione 5.0.6\)](#)
- [Modifiche dei parametri di Redis 5.0.3](#)
- [Redis security](#)

SIC 4: come garantire la crittografia dei dati a riposo con ElastiCache

Introduzione della domanda: sebbene ElastiCache per Redis sia un archivio di dati in memoria, è possibile crittografare tutti i dati che possono essere conservati (nello spazio di archiviazione) come parte delle operazioni standard del cluster. Ad esempio i backup pianificati e manuali scritti su Amazon S3, nonché i dati salvati nello spazio di archiviazione su disco a seguito di operazioni di sincronizzazione e scambio. I tipi di istanza delle famiglie M6g e R6g offrono anche la crittografia in memoria sempre attiva.

Vantaggio della domanda: ElastiCache per Redis offre una crittografia opzionale dei dati a riposo per aumentare la sicurezza.

- [Obbligatorio] La crittografia dei dati a riposo può essere abilitata su un cluster ElastiCache (gruppo di replica) solo al momento della creazione. Un cluster esistente non può essere modificato per iniziare a crittografare i dati a riposo. Per impostazione predefinita, ElastiCache fornisce e gestisce le chiavi utilizzate nella crittografia a riposo.

[Risorse]:

- [Condizioni di crittografia dei dati inattivi](#)
- [Abilitazione della crittografia dei dati inattivi](#)
- [Best practice] Utilizza i tipi di istanza Amazon EC2 che crittografano i dati mentre sono in memoria (come M6g o R6g). Ove possibile, valuta la possibilità di gestire le chiavi per la crittografia a riposo. Per gli ambienti di sicurezza dei dati più rigorosi è possibile utilizzare AWS Key Management Service (KMS) per gestire autonomamente le chiavi master del cliente (CMK). Grazie all'integrazione di ElastiCache con AWS Key Management Service, puoi creare, impostare e gestire le chiavi utilizzate per la crittografia dei dati a riposo per il cluster ElastiCache per Redis.

[Risorse]:

- [Utilizzo di chiavi gestite dal cliente da AWS Key Management Service](#)
- [AWS Key Management Service](#)
- [AWS KMS concepts](#)

SIC 5: come si crittografano i dati in transito con ElastiCache?

Introduzione della domanda: è un requisito comune per evitare che i dati vengano compromessi durante il transito. Rappresenta i dati all'interno dei componenti di un sistema distribuito, nonché tra client applicativi e nodi del cluster. ElastiCache per Redis supporta questo requisito consentendo

la crittografia dei dati in transito tra client e cluster e tra i nodi del cluster stessi. I tipi di istanza delle famiglie M6g e R6g offrono anche la crittografia in memoria sempre attiva.

Vantaggio della domanda: la crittografia in transito di Amazon ElastiCache è una funzionalità opzionale che ti consente di aumentare la sicurezza dei tuoi dati nei punti più vulnerabili, quando sono in transito da una posizione all'altra.

- [Obbligatorio] La crittografia in transito può essere abilitata solo su un cluster ElastiCache per Redis (gruppo di replica) al momento della creazione. Tieni presente che, a causa dell'elaborazione aggiuntiva richiesta per la crittografia/decrittografia dei dati, l'implementazione della crittografia in transito avrà un certo impatto sulle prestazioni. Per comprendere l'impatto, ti consigliamo di confrontare il carico di lavoro prima e dopo aver abilitato la crittografia in transito.

[Risorse]:

- [Panoramica della crittografia dei dati in transito](#)

SIC 6: come si limita l'accesso alle risorse del piano di controllo (control-plane)?

Introduzione della domanda: le policy IAM e l'ARN consentono controlli di accesso granulari per ElastiCache per Redis, abilitando un controllo più rigoroso per gestire la creazione, la modifica e l'eliminazione dei cluster ElastiCache per Redis.

Vantaggio della domanda: la gestione delle risorse di Amazon ElastiCache, come gruppi di replica, nodi e così via può essere limitata agli account AWS con autorizzazioni specifiche basate sulle policy IAM, migliorando la sicurezza e l'affidabilità delle risorse.

- [Obbligatorio] Gestisci l'accesso alle risorse di Amazon ElastiCache assegnando agli utenti AWS policy AWS Identity and Access Management specifiche che consentono un controllo più preciso sulle azioni che gli account possono eseguire sui cluster.

[Risorse]:

- [Panoramica della gestione delle autorizzazioni di accesso alle risorse di ElastiCache](#)
- [Utilizzo di policy basate su identità \(policy IAM\) per Amazon ElastiCache](#)

SIC 7: come si rileva e si risponde agli eventi di sicurezza?

Introduzione della domanda: ElastiCache, se implementato con il RBAC abilitato, esporta i parametri di CloudWatch per notificare agli utenti gli eventi di sicurezza. Queste metriche aiutano a identificare

i tentativi di autenticazione non riusciti, le chiavi di accesso o l'esecuzione di comandi per i quali la connessione degli utenti con il RBAC non è autorizzata.

Inoltre, le risorse relative a prodotti e servizi AWS aiutano a proteggere il carico di lavoro complessivo automatizzando le implementazioni e registrando tutte le azioni e le modifiche per una revisione o un controllo successivo.

Vantaggio della domanda: monitorando gli eventi, consenti all'organizzazione di rispondere in base a requisiti, policy e procedure. L'automazione del monitoraggio e delle risposte a questi eventi rafforza il livello generale di sicurezza.

- [Obbligatorio] Acquisisci familiarità con i parametri di CloudWatch pubblicati relativi agli errori di autenticazione e autorizzazione del RBAC.
 - `AuthenticationFailures` = Tentativi di autenticazione non riusciti su Redis
 - `KeyAuthorizationFailures` = Tentativi degli utenti di accedere alle chiavi senza autorizzazione non riusciti
 - `CommandAuthorizationFailures` = Tentativi degli utenti di eseguire comandi senza autorizzazione non riusciti

[Risorse]:

- [Parametri di Redis](#)
- [Best practice] Ti consigliamo di configurare avvisi e notifiche su queste metriche e rispondere se necessario.

[Risorse]:

- [Utilizzo degli allarmi di Amazon CloudWatch](#)
- [Best practice] Usa il comando `ACL LOG` di Redis per raccogliere ulteriori dettagli

[Risorse]:

- [ACL LOG](#)
- [Best practice] Acquisisci familiarità con le funzionalità dei prodotti e dei servizi AWS per quanto riguarda il monitoraggio, la registrazione e l'analisi delle implementazioni e degli eventi di ElastiCache

[Risorse]:

- [Registrazione delle chiamate all'API Amazon ElastiCache con AWS CloudTrail](#)
- [elasticache-redis-cluster-automatic-backup-check](#)

- [Monitoraggio dell'utilizzo con i parametri di CloudWatch](#)

Pilastro dell'affidabilità di Amazon ElastiCache Well-Architected Lens

Argomenti

- [AFF 1: in che modo si supportano le implementazioni di architetture ad alta disponibilità?](#)
- [AFF 2: in che modo si raggiungono gli obiettivi del punto di ripristino \(RPO\) con ElastiCache?](#)
- [AFF 3: come si supportano i requisiti di ripristino di emergenza?](#)
- [AFF 4: come si pianificano i failover in modo efficace?](#)
- [AFF 5: i tuoi componenti ElastiCache sono progettati per il dimensionamento?](#)

AFF 1: in che modo si supportano le implementazioni di architetture ad alta disponibilità?

Introduzione della domanda: l'architettura ad alta disponibilità di Amazon ElastiCache ti consente di operare in uno stato resiliente durante gli eventi di disponibilità.

Vantaggio della domanda: l'architettura dei cluster ElastiCache resilienti ai guasti garantisce una maggiore disponibilità per le implementazioni ElastiCache.

- [Obbligatorio] Determina il livello di affidabilità richiesto per il cluster ElastiCache in uso. Carichi di lavoro diversi hanno standard di resilienza diversi, da quelli totalmente effimeri a quelli mission critical. Definisci le esigenze per ogni tipo di ambiente in cui gestisci, ad esempio, sviluppo, test e produzione.

Motore di cache: Memcached ed ElastiCache per Redis

1. Memcached non fornisce alcun meccanismo di replica e viene utilizzato principalmente per carichi di lavoro effimeri.
 2. ElastiCache per Redis offre le funzionalità per l'alta disponibilità descritte di seguito
- [Best practice] Per i carichi di lavoro che richiedono alta disponibilità, usa ElastiCache per Redis in modalità cluster con un minimo di due repliche per partizione, anche per carichi di lavoro con requisiti di velocità di trasmissione effettiva contenuti che richiedono una sola partizione.
 1. Con la modalità cluster abilitata, multi-AZ viene impostato automaticamente.

Multi-AZ riduce al minimo i tempi di inattività eseguendo failover automatici dal nodo primario alle repliche, in caso di manutenzione pianificata o non pianificata, mitigando i guasti delle zone di disponibilità.

2. Per i carichi di lavoro a partizioni, un minimo di tre partizioni garantisce un ripristino più rapido durante gli eventi di failover, poiché il protocollo per i cluster Redis richiede la disponibilità della maggior parte dei nodi primari per raggiungere il quorum.
3. Configura due o più repliche per la disponibilità.

La presenza di due repliche offre una migliore scalabilità di lettura e anche la disponibilità di lettura in scenari in cui una replica è in fase di manutenzione.

4. Usa i tipi di nodi basati su Graviton2 (nodi predefiniti nella maggior parte delle regioni).

Amazon ElastiCache per Redis offre prestazioni ottimizzate per questi nodi. Di conseguenza, si ottengono migliori prestazioni di replica e sincronizzazione, con conseguente maggiore disponibilità complessiva.

5. Monitora e definisci le dimensioni corrette per gestire i picchi di traffico previsti. In caso di carichi elevati, il motore ElastiCache per Redis potrebbe non rispondere, influenzando sulla disponibilità. `BytesUsedForCache` e `DatabaseMemoryUsagePercentage` sono buoni indicatori dell'utilizzo della memoria, mentre `ReplicationLag` è un indicatore dello stato della replica in base alla velocità di scrittura. Puoi utilizzare queste metriche per attivare il dimensionamento dei cluster.
6. Garantisci la resilienza lato client eseguendo test con l'[API di failover prima di un evento di failover di produzione](#).

[Risorse]:

- [Configure Amazon ElastiCache for Redis for higher availability](#)
- [Alta disponibilità utilizzando gruppi di replica](#)

AFF 2: in che modo si raggiungono gli obiettivi del punto di ripristino (RPO) con ElastiCache?

Introduzione della domanda: comprendi l'obiettivo del punto di ripristino del carico di lavoro per prendere decisioni informate sulle strategie di backup e ripristino di ElastiCache.

Vantaggio della domanda: disporre di una strategia RPO in atto può migliorare la continuità aziendale in caso di scenari di ripristino di emergenza. La progettazione delle policy di backup e

ripristino può aiutarti a raggiungere i tuoi obiettivi del punto di ripristino (RPO) per i dati ElastiCache. ElastiCache per Redis offre funzionalità di snapshot archiviate in Amazon S3, insieme a una policy di conservazione configurabile. Questi snapshot vengono acquisiti durante la finestra di backup definita e gestiti automaticamente dal servizio. Se il carico di lavoro richiede una maggiore granularità del backup, hai la possibilità di creare fino a 20 backup manuali al giorno. I backup creati manualmente non hanno una policy di conservazione del servizio e possono essere conservati a tempo indeterminato.

- [Obbligatorio] Comprendi e documenta l'obiettivo del punto di ripristino delle implementazioni ElastiCache.
 - Tieni presente che Memcached non offre processi di backup.
 - Esamina le funzionalità di backup e ripristino di ElastiCache.
- [Best practice] Predisponi di una procedura di comunicazione per il backup del cluster.
 - Avvia i backup manuali in base alle necessità.
 - Esamina le policy di conservazione per i backup automatici.
 - Tieni presente che i backup manuali vengono conservati a tempo indeterminato.
 - Pianifica i backup automatici nei periodi di basso utilizzo.
 - Esegui operazioni di backup su repliche di lettura per ridurre al minimo l'impatto sulle prestazioni del cluster.
- [Positivo] Sfrutta la funzionalità di backup pianificato di ElastiCache per eseguire regolarmente il backup dei dati durante una finestra definita.
 - Esegui periodicamente il test del ripristino dei tuoi backup.
- [Risorse]:
 - [Redis](#)
 - [Backup e ripristino per ElastiCache for Redis](#)
 - [Creazione di backup manuali](#)
 - [Pianificazione di backup automatici](#)
 - [Backup and Restore ElastiCache Redis Clusters](#)

AFF 3: come si supportano i requisiti di ripristino di emergenza?

Introduzione della domanda: il ripristino di emergenza è un aspetto importante di qualsiasi pianificazione del carico di lavoro. ElastiCache per Redis offre diverse opzioni per implementare il ripristino di emergenza in base ai requisiti di resilienza del carico di lavoro. Con Global Datastore di

Amazon ElastiCache per Redis puoi scrivere sul tuo cluster ElastiCache per Redis in una regione e avere i dati disponibili per la lettura da altri due cluster di replica interregionali, consentendo così letture a bassa latenza e ripristino di emergenza tra le regioni.

Vantaggio della domanda: la comprensione e la pianificazione di una varietà di scenari di emergenza possono garantire la continuità aziendale. Le strategie di ripristino di emergenza devono essere bilanciate rispetto ai costi, all'impatto sulle prestazioni e alla potenziale perdita di dati.

- [Obbligatorio] Sviluppa e documenta strategie di ripristino di emergenza per tutti i componenti ElastiCache in base ai requisiti del carico di lavoro. ElastiCache è unico in quanto alcuni casi d'uso sono del tutto effimeri e non richiedono alcuna strategia di ripristino di emergenza, mentre altri sono all'estremità opposta dello spettro e richiedono una strategia di ripristino di emergenza estremamente robusta. Tutte le opzioni devono essere valutate rispetto all'ottimizzazione dei costi: una maggiore resilienza richiede una maggiore quantità di infrastruttura.

Comprendi le opzioni di ripristino di emergenza disponibili a livello regionale e multiregionale.

- Le implementazioni multi-AZ sono consigliate per evitare errori di zone di disponibilità. Assicurati di eseguire l'implementazione con la modalità cluster abilitata nelle architetture multi-AZ, con un minimo di 3 zone di disponibilità.
- Global Datastore è consigliato per proteggersi dagli errori a livello di regione.
- [Best practice] Abilita Global Datastore per i carichi di lavoro che richiedono resilienza a livello di regione.
 - Prepara un piano di failover nella regione secondaria in caso di degrado di quella primaria.
 - Esegui il test del processo di failover multiregione prima di eseguire un failover in produzione.
 - Monitora la metrica `ReplicationLag` per comprendere il potenziale impatto della perdita di dati durante gli eventi di failover.
- [Risorse]:
 - [Limitazione dell'impatto degli errori](#)
 - [Replica tra regioni AWS mediante Global Datastore](#)
 - [Ripristino da un backup con ridimensionamento cluster opzionale](#)
 - [Riduzione dei tempi di inattività in ElastiCache for Redis con Multi-AZ](#)

AFF 4: come si pianificano i failover in modo efficace?

Introduzione della domanda: l'abilitazione di multi-AZ con failover automatici è una best practice di ElastiCache. In alcuni casi, ElastiCache per Redis sostituisce i nodi primari come parte delle operazioni di servizio. Ad esempio nel caso di eventi di manutenzione programmata e nel caso poco probabile di un errore in un nodo o una zona di disponibilità. I failover ottimali si basano su ElastiCache e sulla configurazione della libreria client.

Vantaggio della domanda: seguire le best practice per i failover ElastiCache in combinazione con la libreria client specifica di ElastiCache per Redis consente di ridurre al minimo i potenziali tempi di inattività durante gli eventi di failover.

- [Obbligatorio] Per la modalità cluster disabilitata, utilizza i timeout in modo che i client rilevino se è necessario disconnettersi dal vecchio nodo primario e riconnettersi al nuovo nodo primario, utilizzando l'indirizzo IP dell'endpoint primario aggiornato. Per la modalità cluster abilitata, la libreria client è responsabile del rilevamento delle modifiche nella topologia del cluster sottostante. Questa operazione viene spesso eseguita tramite le impostazioni di configurazione nella libreria client ElastiCache per Redis, che consentono anche di configurare la frequenza e il metodo di aggiornamento. Ogni libreria client offre le proprie impostazioni e maggiori dettagli sono disponibili nella documentazione corrispondente.

[Risorse]:

- [Riduzione dei tempi di inattività in ElastiCache for Redis con Multi-AZ](#)
- Esamina le best practice della tua libreria client ElastiCache per Redis.
- [Obbligatorio] Il successo dei failover dipende da un ambiente di replica integro tra il nodo primario e quello di replica. Esamina e comprendi la natura asincrona della replica Redis e le metriche CloudWatch disponibili per segnalare il ritardo di replica tra il nodo primario e quello di replica. Per i casi d'uso che richiedono una maggiore sicurezza dei dati, utilizza il comando WAIT di Redis per forzare le repliche a riconoscere le scritture prima di rispondere ai client connessi.

[Risorse]:

- [Parametri di Redis](#)
- [Monitoring best practices with Amazon ElastiCache for Redis using Amazon CloudWatch](#)
- [Best practice] Convalida regolarmente la reattività dell'applicazione durante il failover utilizzando l'API Test Failover di ElastiCache.

[Risorse]:

- [Testing Automatic Failover to a Read Replica on Amazon ElastiCache for Redis](#)
- [Test del failover automatico](#)

AFF 5: i tuoi componenti ElastiCache sono progettati per il dimensionamento?

Introduzione della domanda: comprendendo le funzionalità di dimensionamento e le topologie di implementazione disponibili, i componenti ElastiCache possono adattarsi nel tempo per soddisfare i requisiti del carico di lavoro in continuo cambiamento. ElastiCache offre un dimensionamento in 4 direzioni: in/out (orizzontale) e up/down (verticale).

Vantaggio della domanda: seguire le best practice per le implementazioni di ElastiCache offre la massima flessibilità di dimensionamento, oltre a soddisfare il principio Well-Architected del dimensionamento orizzontale per ridurre al minimo l'impatto degli errori.

- [Obbligatorio] Comprendi la differenza tra le topologie modalità cluster abilitata e modalità cluster disabilitata. In quasi tutti i casi è consigliabile eseguire l'implementazione con la modalità cluster abilitata in quanto consente una maggiore scalabilità nel tempo. I componenti in modalità cluster disabilitata sono limitati nella capacità di dimensionarsi orizzontalmente per aggiungere repliche di lettura.
- [Obbligatorio] Determina quando e come dimensionare.
 - Per ulteriori READIOPS: aggiungi repliche
 - Per ulteriori WRITEOPS: aggiungi partizioni (dimensionamento orizzontale)
 - Per ulteriori I/O di rete: utilizza istanze ottimizzate per la rete (dimensionamento verticale)
- [Best practice] Implementa i componenti ElastiCache con la modalità cluster abilitata, preferendo tanti piccoli nodi a pochi nodi grandi. In tal modo limiti efficacemente il raggio di applicazione dell'errore di un nodo.
- [Best practice] Includi le repliche nei cluster per una maggiore reattività durante gli eventi di dimensionamento
- [Positivo] Se la modalità cluster è disabilitata, utilizza le repliche di lettura per aumentare la capacità di lettura complessiva. ElastiCache supporta fino a 5 repliche di lettura in modalità cluster disabilitata, oltre al dimensionamento verticale.
- [Risorse]:
 - [Dimensionamento ElastiCache per i cluster Redis](#)
 - [Dimensionamento verso l'alto online](#)
 - [Dimensionamento ElastiCache per i cluster Memcached](#)

Pilastro dell'efficienza delle prestazioni di Amazon ElastiCache Well-Architected Lens

Il pilastro dell'efficienza delle prestazioni si concentra sull'uso efficiente delle risorse IT e di calcolo. Gli argomenti chiave includono la selezione delle dimensioni e dei tipi corretti per le risorse in base ai requisiti del carico di lavoro, il monitoraggio delle prestazioni e il processo per prendere decisioni informate e mantenere l'efficienza man mano che le esigenze aziendali cambiano.

Argomenti

- [EP 1: come si monitorano le prestazioni del cluster Amazon ElastiCache?](#)
- [EP 2: come si distribuisce il lavoro tra i nodi del cluster ElastiCache?](#)
- [EP 3: come si monitora e si segnala l'efficacia e le prestazioni della cache per i carichi di lavoro con memorizzazione nella cache?](#)
- [EP 4: in che modo il carico di lavoro ottimizza l'uso delle risorse e delle connessioni di rete?](#)
- [EP 5: come si gestisce l'eliminazione e/o l'espulsione delle chiavi?](#)
- [EP 6: come si modellano e si usano i dati in ElastiCache?](#)
- [EP 7: come si registrano i comandi a esecuzione lenta nel cluster Amazon ElastiCache?](#)
- [EP 8: in che modo il dimensionamento automatico aiuta ad aumentare le prestazioni del cluster ElastiCache?](#)

EP 1: come si monitorano le prestazioni del cluster Amazon ElastiCache?

Introduzione della domanda: comprendendo le metriche di monitoraggio esistenti è possibile determinare l'utilizzo corrente. Un monitoraggio adeguato può aiutare a individuare i potenziali ostacoli che influiscono sulle prestazioni del cluster.

Vantaggio della domanda: la comprensione delle metriche associate al cluster può aiutare nella definizione delle tecniche di ottimizzazione volte a conseguire la riduzione della latenza e l'aumento della velocità di trasmissione effettiva.

- [Obbligatorio] Test delle prestazioni di base utilizzando un sottoinsieme del carico di lavoro.
 - È necessario monitorare le prestazioni del carico di lavoro effettivo utilizzando meccanismi come i test di carico.
 - Monitora le metriche di CloudWatch durante l'esecuzione di questi test per comprendere i valori restituiti e stabilire la linea di base delle prestazioni.

- [Best practice] Per i carichi di lavoro di ElastiCache per Redis, rinomina i comandi costosi in termini di calcolo, ad esempio KEYS, per ridurre la possibilità che gli utenti eseguano comandi che bloccano i cluster di produzione.
- I carichi di lavoro di ElastiCache per Redis che eseguono il motore 6.x possono utilizzare il controllo degli accessi basato su ruoli per limitare determinati comandi. L'accesso ai comandi può essere controllato creando utenti e gruppi di utenti con la console AWS o la CLI e quindi associandoli a un cluster ElastiCache per Redis. Quando il controllo degli accessi basato su ruoli è abilitato in Redis 6, è possibile usare "-@dangerous" per impedire all'utente l'esecuzione di comandi costosi come KEYS, MONITOR, SORT e così via.
- Per la versione 5.x del motore, rinomina i comandi utilizzando il parametro `rename-commands` nel gruppo di parametri del cluster Amazon ElastiCache per Redis.
- [Consigliato] Analizza le query lente ed esamina le tecniche di ottimizzazione.
 - Per i carichi di lavoro di ElastiCache per Redis puoi analizzare meglio le tue query usando il log delle query lente. Ad esempio, puoi utilizzare il seguente comando `redis-cli slowlog get 10` per mostrare gli ultimi 10 comandi che hanno superato la soglia di latenza (10 secondi per impostazione predefinita).
 - Alcune query possono essere eseguite in modo più efficiente utilizzando le strutture di dati complesse di ElastiCache per Redis. Ad esempio, per le ricerche con intervalli di numeri, è possibile implementare nell'applicazione semplici indici numerici con i set ordinati. La gestione di questi indici può ridurre le scansioni eseguite sui set e restituire i dati con prestazioni migliori.
 - Per i carichi di lavoro di ElastiCache per Redis, `redis-benchmark` fornisce una semplice interfaccia per testare le prestazioni di diversi comandi utilizzando gli input definiti dall'utente, come il numero di client e la dimensione dei dati.
 - Poiché Memcached supporta solo semplici comandi a livello di chiave, valuta la possibilità di creare altre chiavi come indici per evitare l'iterazione dello spazio delle chiavi per rispondere alle query dei client.
- [Risorse]:
 - [Monitoraggio dell'utilizzo con i parametri di CloudWatch](#)
 - [Monitoraggio dell'utilizzo con i parametri di CloudWatch](#)
 - [Utilizzo degli allarmi di Amazon CloudWatch](#)
 - [Parametri specifici di Redis](#)
 - [SLOWLOG](#)
 - [Redis benchmark](#)

EP 2: come si distribuisce il lavoro tra i nodi del cluster ElastiCache?

Introduzione della domanda: il modo in cui l'applicazione si connette ai nodi Amazon ElastiCache può influire sulle prestazioni e sulla scalabilità del cluster.

Vantaggio della domanda: l'uso corretto dei nodi disponibili nel cluster garantisce la distribuzione del lavoro tra le risorse disponibili. Le seguenti tecniche consentono anche di evitare risorse inutilizzate.

- [Obbligatorio] I client devono connettersi all'endpoint ElastiCache appropriato.
 - Amazon ElastiCache per Redis implementa diversi endpoint in base alla modalità cluster in uso. Se la modalità cluster è abilitata, ElastiCache fornisce un endpoint di configurazione. Se la modalità cluster è disabilitata, ElastiCache fornisce un endpoint primario, in genere utilizzato per le scritture, e un endpoint di lettura per bilanciare le letture tra le repliche. L'implementazione corretta di questi endpoint si traduce in prestazioni migliori e operazioni di dimensionamento più semplici. Evita di connetterti ai singoli endpoint dei nodi a meno che non vi sia un requisito specifico in tal senso.
 - Per i cluster Memcached con più nodi, ElastiCache fornisce un endpoint di configurazione che abilita l'individuazione automatica. Ti consigliamo di utilizzare un algoritmo di hash per distribuire il lavoro in modo uniforme tra i nodi di cache. Molte librerie client Memcached implementano l'hash in modo coerente. Consulta la documentazione della libreria che utilizzi per verificare se supporta l'hashing coerente e ottenere informazioni su come implementarlo. Ulteriori informazioni sull'implementazione di queste funzionalità sono disponibili [qui](#).
- [Consigliato] Sfrutta la modalità cluster ElastiCache per Redis abilitata per migliorare la scalabilità.
 - I cluster ElastiCache per Redis (in modalità cluster abilitata) supportano le [operazioni di dimensionamento online](#) (orizzontale/verticale) per consentire di distribuire i dati in modo dinamico tra le partizioni. L'utilizzo dell'endpoint di configurazione garantisce che i client che supportano il cluster possano adattarsi ai cambiamenti nella topologia del cluster.
 - Puoi anche ribilanciare il cluster spostando gli hash slot tra le partizioni disponibili nel cluster ElastiCache per Redis (modalità cluster abilitata). In tal modo il lavoro viene distribuito in modo più efficiente tra le partizioni disponibili.
- [Consigliato] Implementa una strategia per identificare e correggere le chiavi hot nel tuo carico di lavoro.
 - Considera l'impatto delle strutture di dati Redis multidimensionali come elenchi, flussi, set e così via. Queste strutture di dati sono archiviate in singole chiavi Redis che risiedono in un solo nodo. Una chiave multidimensionale molto grande può utilizzare più memoria e capacità di rete rispetto

ad altri tipi di dati e causare un uso sproporzionato del nodo. Se possibile, progetta il tuo carico di lavoro in modo da distribuire l'accesso ai dati tra molte chiavi discrete.

- Le chiavi hot del carico di lavoro possono influire sulle prestazioni del nodo in uso. Nei carichi di lavoro di ElastiCache per Redis, è possibile rilevare le chiavi hot utilizzando `redis-cli --hotkeys` se è in uso una policy LFU per la memoria massima.
- Prendi in considerazione la replica delle chiavi hot su più nodi per distribuire l'accesso in modo più uniforme. Con questo approccio il client scrive su più nodi primari (il nodo Redis non fornisce questa funzionalità) e mantiene un elenco di nomi di chiavi da leggere, oltre al nome della chiave originale.
- ElastiCache per Redis versione 6 supporta la [memorizzazione nella cache lato client](#) assistita dal server. In tal modo le applicazioni possono attendere le modifiche apportate alle chiavi prima di effettuare le chiamate di rete a ElastiCache.
- [Risorse]:
 - [Configure Amazon ElastiCache for Redis for higher availability](#)
 - [Individuazione degli endpoint di connessione](#)
 - [Best practice per il bilanciamento del carico](#)
 - [Reshard online e ribilanciamento della partizione per Redis \(modalità cluster abilitata\)](#)
 - [Client-side caching in Redis](#)

EP 3: come si monitora e si segnala l'efficacia e le prestazioni della cache per i carichi di lavoro con memorizzazione nella cache?

Introduzione della domanda: la memorizzazione nella cache è un carico di lavoro comune in ElastiCache ed è importante capire come gestire l'efficacia e le prestazioni della cache.

Vantaggio della domanda: l'applicazione potrebbe mostrare segni di rallentamento delle prestazioni. La capacità di utilizzare metriche specifiche della cache per decidere come aumentare le prestazioni delle app è fondamentale per il carico di lavoro della cache.

- [Obbligatorio] Misura e monitora nel tempo il rapporto dei riscontri della cache. L'efficienza della cache è determinata dal "rapporto di riscontri della cache". Il rapporto di riscontri della cache è definito dal totale dei riscontri delle chiavi diviso per il totale dei riscontri e dei mancati riscontri. Più il rapporto è vicino a 1, più efficace è la cache. Un basso rapporto di riscontri della cache è causato dal volume di mancati riscontri della cache. I mancati riscontri della cache si verificano quando la chiave richiesta non viene trovata nella cache. Una chiave non è nella cache perché è

stata rimossa o eliminata, è scaduta o non è mai esistita. Determina il motivo per cui le chiavi non sono nella cache e sviluppa le strategie appropriate per averle nella cache.

[Risorse]:

- [Parametri di Redis](#)
- [Obbligatorio] Misura e raccogli le prestazioni della cache delle applicazioni insieme ai valori di latenza e utilizzo della CPU per capire se è necessario apportare modifiche al time-to-live o ad altri componenti dell'applicazione. ElastiCache fornisce una serie di metriche di CloudWatch per le latenze aggregate per ogni struttura di dati. Queste metriche per le latenze sono calcolate utilizzando le statistiche commandstats del comando INFO di ElastiCache per Redis e non includono i tempi della rete e di I/O, ma solo il tempo impiegato da ElastiCache per Redis per elaborare le operazioni.

[Risorse]:

- [Parametri di Redis](#)
- [Monitoring best practices with Amazon ElastiCache for Redis using Amazon CloudWatch](#)
- [Best practice] Scegli la strategia di memorizzazione nella cache giusta per le tue esigenze. Un basso rapporto di riscontri della cache è causato dal volume di mancati riscontri della cache. Se il carico di lavoro è progettato per avere un volume ridotto di mancati riscontri della cache (come la comunicazione in tempo reale), è consigliabile esaminare le strategie di memorizzazione nella cache e applicare le risoluzioni più appropriate per il carico di lavoro, ad esempio la strumentazione delle query per misurare la memoria e le prestazioni. Le strategie effettive utilizzate per implementare il popolamento e la gestione della cache dipende dai dati del client che desideri memorizzare nella cache e dai modelli di accesso a tali dati. Ad esempio, è improbabile che utilizzi in un'applicazione di streaming la stessa strategia per i suggerimenti personalizzati e per le notizie più interessanti.

[Risorse]:

- [Strategie di caching](#)
- [Caching Best Practices](#)
- [Performance at Scale with Amazon ElastiCache Whitepaper](#)

EP 4: in che modo il carico di lavoro ottimizza l'uso delle risorse e delle connessioni di rete?

Introduzione della domanda: ElastiCache per Redis e Memcached sono supportati da molti client di applicazioni e le implementazioni possono variare. È necessario comprendere la gestione della rete e delle connessioni in uso per analizzare il potenziale impatto sulle prestazioni.

Vantaggio della domanda: l'uso corretto delle risorse di rete può migliorare l'efficienza delle prestazioni del cluster. I seguenti suggerimenti possono ridurre le richieste di rete e migliorare la latenza e la velocità di trasmissione effettiva del cluster.

- [Obbligatorio] Gestisci in modo proattivo le connessioni al tuo cluster ElastiCache.
 - Il pool di connessioni dell'applicazione riduce la quantità di sovraccarico sul cluster creato dall'apertura e dalla chiusura delle connessioni. Monitora il comportamento delle connessioni in Amazon CloudWatch utilizzando `CurConnections` e `NewConnections`.
 - Evita di perdere le connessioni client chiudendole correttamente, quando opportuno. Le strategie di gestione delle connessioni includono la corretta chiusura delle connessioni non in uso e l'impostazione del timeout delle connessioni.
 - Per i carichi di lavoro Memcached, esiste una quantità di memoria configurabile riservata alla gestione delle connessioni chiamate, `memcached_connections_overhead`.
- [Consigliato] Comprimi gli oggetti di grandi dimensioni per ridurre la memoria e migliorare la velocità di trasmissione effettiva di rete.
 - La compressione dei dati può ridurre la velocità di trasmissione effettiva di rete richiesta (Gbps), ma aumenta la quantità di lavoro dell'applicazione per comprimere e decomprimere i dati.
 - La compressione riduce anche la quantità di memoria consumata dalle chiavi.
 - In base alle esigenze della tua applicazione, considera i compromessi tra rapporto di compressione e velocità di compressione.
- [Risorse]:
 - [Global Datastore di Amazon ElastiCache per Redis](#)
 - [Parametri Memcached specifici](#)
 - [Amazon ElastiCache for Redis 5.0.3 migliora la gestione I/O per incrementare le prestazioni](#)
 - [Parametri di Redis](#)
 - [Configure Amazon ElastiCache for Redis for higher availability](#)

EP 5: come si gestisce l'eliminazione e/o l'espulsione delle chiavi?

Introduzione della domanda: i carichi di lavoro hanno requisiti e comportamenti previsti diversi quando un nodo del cluster si avvicina ai limiti di consumo della memoria. Amazon ElastiCache per Redis prevede policy diverse per la gestione di queste situazioni.

Vantaggio della domanda: la corretta gestione della memoria disponibile e la comprensione delle policy di espulsione contribuiscono a garantire la consapevolezza del comportamento del cluster quando i limiti di memoria delle istanze vengono superati.

- [Obbligatorio] Strumenta l'accesso ai dati per valutare quale policy applicare. Identifica una policy per la memoria massima appropriata per controllare se e come vengono eseguite le espulsioni sul cluster.
 - L'espulsione si verifica quando viene consumata la memoria massima del cluster e c'è una policy che consente l'operazione. Il comportamento del cluster in questa situazione dipende dalla policy di espulsione specificata. Questa policy può essere gestita utilizzando `maxmemory-policy` nel gruppo di parametri del cluster ElastiCache per Redis.
 - La policy predefinita `volatile-lru` libera la memoria espellendo le chiavi con un tempo di scadenza impostato (valore TTL). Le policy per l'utilizzo meno frequente (LFU) e meno recente (LRU) rimuovono le chiavi in base all'utilizzo.
 - Per i carichi di lavoro Memcached, esiste una policy predefinita per LRU che controlla le espulsioni su ogni nodo. Il numero di espulsioni sul cluster Amazon ElastiCache può essere monitorato utilizzando la metrica delle espulsioni in Amazon CloudWatch.
- [Best practice] Standardizza il comportamento di eliminazione per controllare l'impatto sulle prestazioni del cluster ed evitare imprevisti colli di bottiglia delle prestazioni.
 - Quando si rimuovono esplicitamente le chiavi dal cluster nei carichi di lavoro di ElastiCache per Redis, `UNLINK` è equivalente a `DEL`, rimuove le chiavi specificate. Tuttavia, il comando esegue il recupero della memoria effettiva in un thread diverso, quindi non si blocca, contrariamente a `DEL`. La rimozione effettiva avviene successivamente in modo asincrono.
 - Per i carichi di lavoro di ElastiCache per Redis 6.x, il comportamento del comando `DEL` può essere modificato nel gruppo di parametri utilizzando il parametro `lazyfree-lazy-user-del`.
- [Risorse]:
 - [Configurazione dei parametri di motore con i gruppi di parametri](#)
 - [UNLINK](#)
 - [Gestione finanziaria del cloud con AWS](#)

EP 6: come si modellano e si usano i dati in ElastiCache?

Introduzione della domanda: ElastiCache dipende fortemente dall'applicazione per le strutture di dati e il modello di dati utilizzati, ma deve anche considerare l'archivio dati sottostante (se presente). Esamina le strutture di dati di ElastiCache per Redis disponibili e assicurati di utilizzare quelle più appropriate per le tue esigenze.

Vantaggio della domanda: la modellazione dei dati in ElastiCache ha diversi livelli, tra cui il caso d'uso dell'applicazione, i tipi di dati e le relazioni tra gli elementi di dati. Inoltre, ogni tipo di dati e comando di ElastiCache per Redis ha proprie caratteristiche prestazionali ben documentate.

- [Best practice] Una best practice consiste nel ridurre la sovrascrittura involontaria dei dati. Utilizza una convenzione di denominazione che riduca al minimo la sovrapposizione dei nomi delle chiavi. La convenzione di denominazione delle strutture di dati utilizza il metodo gerarchico APPNAME:CONTEXT:ID, ad esempio ORDER-APP:CUSTOMER:123.

[Risorse]:

- [Key naming](#)
- [Best practice] I comandi di ElastiCache per Redis hanno una complessità temporale definita dalla notazione Big O. La complessità temporale di un comando è la rappresentazione algoritmica/matematica del suo impatto. Quando si introduce un nuovo tipo di dati nell'applicazione, è necessario esaminare attentamente la complessità temporale dei relativi comandi. I comandi con la complessità temporale $O(1)$ sono costanti nel tempo e non dipendono dalla dimensione dell'input, mentre i comandi con la complessità temporale $O(N)$ sono lineari nel tempo e sono soggetti alla dimensione dell'input. A causa della progettazione a thread singolo di ElastiCache per Redis, un elevato volume di operazioni ad alta complessità temporale comporta una riduzione delle prestazioni e potenziali timeout operativi.

[Risorse]:

- [Comandi](#)
- [Best practice] Usa le API per ottenere la visibilità della GUI nel modello di dati del cluster.

[Risorse]:

- [Redis Commander](#)
- [Redis Browser](#)
- [Redsmin](#)

EP 7: come si registrano i comandi a esecuzione lenta nel cluster Amazon ElastiCache?

Introduzione della domanda: vantaggi dell'ottimizzazione delle prestazioni attraverso l'acquisizione, l'aggregazione e la notifica di comandi di lunga durata. Comprendendo il tempo necessario per l'esecuzione dei comandi, è possibile determinare quali comandi comportano prestazioni scadenti e quali impediscono al motore di funzionare in modo ottimale. Amazon ElastiCache per Redis include anche la possibilità di inoltrare queste informazioni ad Amazon CloudWatch o Amazon Kinesis Data Firehose.

Vantaggio della domanda: l'accesso a una posizione permanente dedicata e l'invio di eventi di notifica per i comandi lenti possono aiutare a eseguire un'analisi dettagliata delle prestazioni e possono essere utilizzati per attivare eventi automatizzati.

- [Obbligatorio] Amazon ElastiCache per Redis esegue la versione 6.0 o successiva del motore, il gruppo di parametri è correttamente configurato e la registrazione SLOWLOG è abilitata sul cluster.
 - I parametri richiesti sono disponibili solo quando la compatibilità della versione del motore è impostata su Redis versione 6.0 o superiore.
 - La registrazione SLOWLOG si verifica quando l'esecuzione di un comando sul server richiede più tempo del valore specificato. Il comportamento del cluster dipende dai parametri `slowlog-log-slower-than` e `slowlog-max-len` del gruppo di parametri associato.
 - Le modifiche diventano effettive immediatamente.
- [Best practice] Utilizza le funzionalità di CloudWatch o Kinesis Data Firehose.
 - Usa le funzionalità di filtro e allarme di CloudWatch, CloudWatch Logs Insights e Amazon Simple Notification Services per eseguire il monitoraggio delle prestazioni e la notifica degli eventi.
 - Usa le funzionalità di streaming di Kinesis Data Firehose per archiviare i log SLOWLOG in uno spazio di archiviazione permanente o per attivare l'ottimizzazione automatica dei parametri del cluster.
 - Determina se si adatta meglio alle tue esigenze il formato JSON o TEXT normale.
 - Fornisci le autorizzazioni IAM per la pubblicazione su CloudWatch o Kinesis Data Firehose.
- [Consigliato] Configura `slowlog-log-slower-than` con un valore diverso da quello predefinito.
 - Questo parametro determina per quanto tempo un comando può essere eseguito nel motore Redis prima che venga registrato come comando a esecuzione lenta. Il valore predefinito è 10.000 microsecondi (10 millisecondi). Il valore predefinito potrebbe essere troppo alto per alcuni carichi di lavoro.

- Determina il valore più appropriato per il tuo carico di lavoro in base alle esigenze delle applicazioni e ai risultati dei test, tuttavia considera che un valore troppo basso può generare un volume eccessivo di dati.
- [Consigliato] Lascia `slowlog-max-len` impostato sul valore predefinito.
- Questo parametro determina il numero massimo di comandi a esecuzione lenta che possono essere acquisiti nella memoria Redis in un dato momento. Un valore pari a 0 disabilita l'acquisizione. Più alto è il valore, più elementi verranno archiviati in memoria, riducendo la possibilità che informazioni importanti vengano espulse prima di essere esaminate. Il valore predefinito è 128.
- Il valore predefinito è appropriato per la maggior parte dei carichi di lavoro. Se è necessario analizzare i dati in una finestra temporale estesa dalla CLI di Redis tramite il comando `SLOWLOG`, valuta la possibilità di aumentare questo valore. In tal modo più comandi possono rimanere nella memoria Redis.

I dati `SLOWLOG` inviati a CloudWatch Logs o Kinesis Data Firehose vengono memorizzati in modo persistente e possono essere analizzati all'esterno del sistema ElastiCache, riducendo la necessità di archiviare un elevato numero di comandi a esecuzione lenta nella memoria Redis.

- [Risorse]:
 - [How do I turn on Redis Slow log in an ElastiCache for Redis cache cluster?](#)
 - [Consegna dei registri](#)
 - [Parametri specifici di Redis](#)
 - <https://aws.amazon.com/cloudwatch/> Amazon CloudWatch
 - [Amazon Kinesis Data Firehose](#)

EP 8: in che modo il dimensionamento automatico aiuta ad aumentare le prestazioni del cluster ElastiCache?

Introduzione della domanda: implementando la funzionalità di dimensionamento automatico di Redis, i componenti ElastiCache possono adattarsi nel tempo per aumentare o diminuire automaticamente le partizioni o le repliche desiderate. Questa operazione può essere eseguita implementando il monitoraggio degli obiettivi o la policy di dimensionamento pianificata.

Vantaggio della domanda: la comprensione e la pianificazione dei picchi del carico di lavoro possono garantire prestazioni della cache migliorate e continuità aziendale. Il dimensionamento automatico

di ElastiCache per Redis monitora continuamente l'utilizzo della memoria CPU per garantire che il cluster funzioni ai livelli di prestazioni desiderati.

- [Obbligatorio] Quando si avvia un cluster per ElastiCache per Redis:
 1. Assicurati che la modalità cluster sia abilitata
 2. Assicurati che l'istanza faccia parte della famiglia dei tipi e delle dimensioni che supportano il dimensionamento automatico
 3. Assicurati che il cluster non sia in esecuzione in datastore globali, Outpost o zone locali

[Risorse]:

- [Ridimensionamento dei cluster in Redis \(modalità cluster attivata\)](#)
- [Utilizzo di Auto Scaling con i shard](#)
- [Utilizzo di Auto Scaling con le repliche](#)
- [Best practice] Determina se il tuo carico di lavoro è pesante in lettura o in scrittura quando definisci la policy di dimensionamento. Per le prestazioni ottimali, utilizza una sola metrica di monitoraggio. Ti consigliamo di evitare l'uso di più policy per una dimensione perché le policy di dimensionamento automatico si dimensionano quando l'obiettivo viene raggiunto, ma si ridimensionano solo quando tutte le policy di monitoraggio degli obiettivi sono pronte per farlo.

[Risorse]:

- [Policy di Auto Scaling](#)
- [Definizione di una policy di dimensionamento](#)
- [Best practice] Il monitoraggio delle prestazioni nel tempo può aiutarti a rilevare i cambiamenti del carico di lavoro che rimarrebbero inosservati se monitorati solo in un determinato momento. Puoi analizzare le metriche CloudWatch corrispondenti per l'utilizzo del cluster in un periodo di quattro settimane per determinare la soglia del valore dell'obiettivo. Se non sei ancora certo del valore da scegliere, ti consigliamo di iniziare con il valore predefinito minimo supportato della metrica.

[Risorse]:

- [Monitoraggio dell'utilizzo con i parametri di CloudWatch](#)
- [Consigliato] Consigliamo di testare l'applicazione con i carichi di lavoro minimi e massimi previsti, per identificare il numero esatto di partizioni/repliche necessarie al cluster per sviluppare le policy di dimensionamento e contenere i problemi di disponibilità.

[Risorse]:

- [Registrazione un target scalabile](#)

- [Registrazione un target scalabile](#)

Pilastro dell'ottimizzazione dei costi di Amazon ElastiCache Well-Architected Lens

Il pilastro dell'ottimizzazione dei costi si concentra sull'evitare costi inutili. Gli argomenti chiave includono la comprensione e il controllo delle spese, la selezione del tipo di nodo più appropriato (utilizza istanze che supportano la suddivisione dei dati su più livelli in base alle esigenze del carico di lavoro), il numero corretto dei tipi di risorse (repliche lette), l'analisi delle spese nel tempo e il dimensionamento per soddisfare le esigenze aziendali senza costi eccessivi.

Argomenti

- [COSTI 1: come si identificano e si monitorano i costi associati alle risorse ElastiCache? Come si sviluppano meccanismi per consentire agli utenti di creare, gestire ed eliminare le risorse create?](#)
- [COSTI 2: come si utilizzano gli strumenti di monitoraggio continuo per ottimizzare i costi associati alle risorse di ElastiCache?](#)
- [COSTI 3: è necessario utilizzare un tipo di istanza che supporti la suddivisione dei dati in più livelli? Quali sono i vantaggi di una suddivisione dei dati in più livelli? Quando non utilizzare le istanze di suddivisione dei dati in più livelli?](#)

COSTI 1: come si identificano e si monitorano i costi associati alle risorse ElastiCache? Come si sviluppano meccanismi per consentire agli utenti di creare, gestire ed eliminare le risorse create?

Introduzione della domanda: la comprensione delle metriche di costo richiede la partecipazione e la collaborazione di più team: ingegneria del software, gestione dei dati, proprietari dei prodotti, finanza e leadership. L'identificazione dei principali fattori di costo richiede che tutte le parti coinvolte comprendano le leve di controllo dell'utilizzo dei servizi e i compromessi di gestione dei costi che spesso è la differenza fondamentale tra le attività di ottimizzazione dei costi efficaci e meno efficaci. Assicurarti di disporre di processi e strumenti per tenere traccia delle risorse create durante lo sviluppo, la produzione e il ritiro e gestire i costi associati a ElastiCache.

Vantaggio della domanda: il monitoraggio continuo di tutti i costi associati al carico di lavoro richiede una conoscenza approfondita dell'architettura che include ElastiCache come componente. Inoltre, è necessario disporre di un piano di gestione dei costi per raccogliere e confrontare l'utilizzo rispetto al budget.

- [Obbligatorio] Istituisce un Centro di eccellenza del Cloud (CCoE) con una carta costitutiva per definire, tracciare e agire in base alle metriche relative all'utilizzo di ElastiCache da parte delle organizzazioni. Se un CCoE esiste e funziona, assicurati che possa leggere e tenere traccia dei costi associati a ElastiCache. Quando vengono create le risorse, utilizza i ruoli e le policy IAM per verificare che solo i team e i gruppi specifici possano creare istanze delle risorse. In tal modo i costi sono associati ai risultati aziendali e viene stabilita una chiara linea di responsabilità, dal punto di vista dei costi.
 1. Il CCoE deve identificare, definire e pubblicare metriche di costo aggiornate regolarmente, su base mensile, sull'utilizzo chiave di ElastiCache per i dati di categoria come:
 - a. Tipi di nodi utilizzati e relativi attributi: istanze standard e ottimizzate per la memoria, istanze on demand e riservate, regioni e zone di disponibilità
 - b. Tipi di ambienti: gratuito, sviluppo, test e produzione
 - c. Strategie di backup, archiviazione e conservazione
 - d. Trasferimento di dati all'interno e tra le regioni
 - e. Istanze in esecuzione su Amazon Outposts
 2. Il CCoE è composto da un team interfunzionale con rappresentanze non esclusive dei team di software engineering, gestione dei dati, team di prodotto, finanza e team dirigenziali dell'organizzazione.

[Risorse]:

- [Create a Cloud Center of Excellence](#)
- [Prezzi di Amazon ElastiCache](#)
- [Obbligatorio] Utilizza i tag di allocazione dei costi per tenere traccia dei costi con un basso livello di granularità. Usa Gestione costi AWS per visualizzare, comprendere e gestire i costi e l'utilizzo di AWS nel tempo.
 1. Utilizza i tag per organizzare le tue risorse e tenere traccia dei costi di AWS a livello dettagliato. Dopo aver attivato i tag di allocazione dei costi, AWS li utilizza per organizzare i costi delle risorse nel proprio report di allocazione costi, per semplificare la categorizzazione e il monitoraggio dei costi di AWS. AWS offre due tipi di tag di allocazione dei costi, tag generati da AWS e tag definiti dall'utente. AWS definisce, crea e applica i tag generati da AWS per tuo conto, mentre tu definisci, crei e applichi i tag definiti dall'utente. È necessario attivare entrambi i tipi di tag separatamente per poterli visualizzare in Gestione dei costi o in un report di allocazione dei costi.

2. Per organizzare le fatture AWS al fine di riflettere la struttura dei costi, puoi utilizzare i tag di allocazione dei costi. Quando aggiungi i tag di allocazione dei costi alle tue risorse in Amazon ElastiCache, puoi tenere traccia dei costi raggruppando le spese nelle fatture mediante i valori dei tag della risorsa. Puoi anche combinare i tag per monitorare i costi con un livello di dettagli maggiore.

[Risorse]:

- [Utilizzo dei tag per l'allocazione dei costi AWS](#)
 - [Monitoraggio dei costi con i tag di allocazione dei costi](#)
 - [AWS Cost Explorer](#)
- [Best practice] Collega i costi di ElastiCache alle metriche che riguardano l'intera organizzazione.
 1. Prendi in considerazione le metriche aziendali e le metriche operative come la latenza: quali concetti del tuo modello di business sono comprensibili in tutti i ruoli? Le metriche devono essere comprensibili per il maggior numero possibile di ruoli all'interno dell'organizzazione.
 2. Ad esempio, utenti serviti simultaneamente, latenza massima e media per operazione e utente, punteggi di coinvolgimento degli utenti, percentuali di reso degli utenti/settimana, durata della sessione/utente, tasso di abbandono, percentuale di riscontri nella cache e chiavi monitorate

[Risorse]:

- [Monitoraggio dell'utilizzo con i parametri di CloudWatch](#)
- [Positivo] Mantieni una visibilità operativa e dell'architettura aggiornata su metriche e costi dell'intero carico di lavoro che utilizza ElastiCache.
 1. Comprendendo l'intero ecosistema di soluzioni, ElastiCache tende a far parte di un ecosistema completo di servizi AWS nel set tecnologico, da client a gateway API, Redshift e QuickSight per gli strumenti di reporting (ad esempio).
 2. Mappa i componenti della soluzione (client, connessioni, sicurezza, operazioni in memoria, archiviazione, automazione delle risorse, accesso e gestione dei dati) nel tuo diagramma di architettura. Ogni livello si collega all'intera soluzione e ha proprie esigenze e funzionalità che aumentano e/o aiutano a gestire il costo complessivo.
 3. Il diagramma deve includere l'uso di calcolo, rete, archiviazione, policy del ciclo di vita, raccolta di metriche e gli elementi operativi e funzionali di ElastiCache dell'applicazione
 4. È probabile che i requisiti del carico di lavoro si evolvano nel tempo ed è essenziale continuare a mantenere e documentare la comprensione dei componenti sottostanti e dei propri obiettivi funzionali primari per rimanere proattivi nella gestione dei costi del carico di lavoro.

5. Il supporto esecutivo per visibilità, responsabilità, priorità e risorse è fondamentale per disporre di una strategia di gestione dei costi efficace per ElastiCache.

COSTI 2: come si utilizzano gli strumenti di monitoraggio continuo per ottimizzare i costi associati alle risorse di ElastiCache?

Introduzione della domanda: è necessario puntare a un giusto equilibrio tra i costi di ElastiCache e le metriche delle prestazioni delle applicazioni. Amazon CloudWatch offre visibilità sui principali parametri operativi che possono aiutarti a valutare se le risorse di ElastiCache sono sovrautilizzate o meno, rispetto alle tue esigenze. Dal punto di vista dell'ottimizzazione dei costi, è necessario comprendere quando il provisioning è eccessivo ed essere in grado di sviluppare meccanismi appropriati per ridimensionare le risorse di ElastiCache rispettando al contempo le esigenze operative, di disponibilità, resilienza e prestazioni.

Vantaggio della domanda: in un ambiente ideale, hai messo a disposizione risorse sufficienti per soddisfare le esigenze operative del tuo carico di lavoro e non disponi di risorse sottoutilizzate che possono portare costi non ottimali. È necessario essere in grado di identificare ed evitare di utilizzare risorse di ElastiCache sovradimensionate per lunghi periodi di tempo.

- [Obbligatorio] Usa CloudWatch per monitorare i tuoi cluster ElastiCache e analizzare il modo in cui queste metriche si riferiscono alle dashboard di Esploratore dei costi AWS
 1. ElastiCache fornisce parametri a livello di host (ad esempio, utilizzo della CPU) e parametri che sono specifici del software del motore di cache (ad esempio, cache gets e cache misses). Questi parametri vengono misurati e pubblicati per ogni nodo di cache in intervalli di 60 secondi.
 2. Le metriche delle prestazioni di ElastiCache (CPUUtilization, EngineUtilization, SwapUsage, CurrConnections ed Evictions) possono indicare che è necessario il dimensionamento verticale (utilizza tipi di nodi di cache più grandi/più piccoli) o il dimensionamento orizzontale (aggiungi più/meno partizioni). Comprendi le implicazioni in termini di costi delle decisioni di dimensionamento creando una matrice di playbook che stimi i costi aggiuntivi e i tempi minimi e massimi necessari per raggiungere le soglie di prestazioni delle applicazioni.

[Risorse]:

- [Monitoraggio dell'utilizzo con i parametri di CloudWatch](#)
- [Quali parametri è opportuno monitorare?](#)
- [Prezzi di Amazon ElastiCache](#)
- [Obbligatorio] Comprendi e documenta la tua strategia di backup e le implicazioni in termini di costi.

1. Con ElastiCache, i backup vengono archiviati in Amazon S3, che fornisce uno spazio di archiviazione durevole. È necessario comprendere le implicazioni in termini di costi in relazione alla capacità di ripristino in caso di guasti.
2. Abilita i backup automatici che eliminano i file di backup che hanno superato il limite di conservazione.

[Risorse]:

- [Pianificazione di backup automatici](#)
 - [Prezzi di Amazon S3](#)
- [Best practice] Usa i nodi riservati per le tue istanze come strategia deliberata per gestire i costi dei carichi di lavoro che sono ben appresi e documentati. I nodi riservati vengono addebitati un costo anticipato che dipende dal tipo di nodo e dalla durata della prenotazione, ovvero uno o tre anni. Tale addebito è molto inferiore all'addebito orario di utilizzo previsto in caso di nodi on demand.
 1. Potrebbe essere necessario gestire i cluster ElastiCache utilizzando nodi on demand fino a quando non avrai raccolto dati sufficienti per stimare i requisiti delle istanze riservate. Pianifica e documenta le risorse necessarie per soddisfare le tue esigenze e confronta i costi previsti tra i diversi tipi di istanza (on demand o riservata)
 2. Valuta regolarmente i nuovi tipi di nodi di cache disponibili e se ha senso, dal punto di vista dei costi e delle metriche operative, migrare il parco istanze a nuovi tipi di nodi di cache

COSTI 3: è necessario utilizzare un tipo di istanza che supporti la suddivisione dei dati in più livelli? Quali sono i vantaggi di una suddivisione dei dati in più livelli? Quando non utilizzare le istanze di suddivisione dei dati in più livelli?

Introduzione della domanda: la selezione del tipo di istanza appropriato può avere non solo un impatto sulle prestazioni e sul livello di servizio, ma anche un impatto finanziario. I tipi di istanza hanno costi diversi. La selezione di uno o più tipi di istanze di grandi dimensioni in grado di soddisfare tutte le esigenze di archiviazione in memoria potrebbe essere una decisione naturale. Tuttavia, potrebbe avere un impatto significativo sui costi man mano che il progetto matura. Per garantire la selezione del tipo di istanza corretto è necessario un esame periodico del tempo di inattività degli oggetti ElastiCache.

Vantaggio della domanda: una chiara comprensione dell'impatto dei vari tipi di istanze sui costi attuali e futuri. Le modifiche marginali o periodiche del carico di lavoro non devono causare variazioni sproporzionate dei costi. Se il carico di lavoro lo consente, i tipi di istanze che supportano la suddivisione dei dati in più livelli offrono un prezzo migliore per l'archiviazione disponibile. Grazie allo

spazio di archiviazione SSD disponibile per istanza, le istanze con suddivisione dei dati in più livelli supportano una capacità totale di dati per istanza molto più elevata.

- [Obbligatorio] Comprendi i limiti delle istanze con suddivisione dei dati in più livelli
 1. Disponibile solo per i cluster ElastiCache per Redis.
 2. Solo tipi di istanza limitati supportano la suddivisione dei dati in più livelli.
 3. Solo ElastiCache per Redis versione 6.2 e successive è supportato.
 4. Gli oggetti di grandi dimensioni non vengono sostituiti in SSD. Gli oggetti superiori a 128 MiB vengono conservati in memoria.

[Risorse]:

- [Tiering di dati](#)
- [Prezzi di Amazon ElastiCache](#)
- [Obbligatorio] Scopri la percentuale del database che viene regolarmente utilizzata dal carico di lavoro.
 1. Le istanze con suddivisione dei dati in più livelli sono ideali per i carichi di lavoro che spesso accedono a una piccola parte del set di dati complessivo ma richiedono comunque un accesso rapido ai dati rimanenti. In altre parole, il rapporto tra dati da "ad accesso frequente" ad "a caldo" è di circa 20:80.
 2. Sviluppa il monitoraggio a livello di cluster dei tempi di inattività degli oggetti.
 3. Le grandi implementazioni di oltre 500 GB di dati sono buone candidate
- [Obbligatorio] Tieni presente che le istanze con suddivisione dei dati in più livelli non sono opzionali per determinati carichi di lavoro.
 1. L'accesso agli oggetti usati meno frequentemente comporta un costo in termini di prestazioni ridotto, poiché questi vengono sostituiti nel SSD locale. Se la tua applicazione è sensibile ai tempi di risposta, verifica l'impatto sul carico di lavoro.
 2. Non adatto per cache che memorizzano principalmente oggetti di grandi dimensioni superiori a 128 MiB.

[Risorse]:

- [Limitazioni](#)
- [Best practice] I tipi di istanze riservate supportano la suddivisione dei dati in più livelli. Ciò garantisce il costo più basso in termini di quantità di archiviazione di dati per istanza.

1. Potrebbe essere necessario gestire i cluster ElastiCache utilizzando istanze con suddivisione dei dati in più livelli non basate sui dati fino a quando non si ha una migliore comprensione dei requisiti.
2. Analizza il modello di utilizzo dei dati dei cluster ElastiCache.
3. Crea un processo automatico che raccolga periodicamente i tempi di inattività degli oggetti.
4. Se noti che un'alta percentuale (circa l'80%) di oggetti è inattiva per un periodo di tempo ritenuto appropriato per il tuo carico di lavoro, documenta i risultati e suggerisci di migrare il cluster su istanze che supportano la suddivisione dei dati in più livelli.
5. Valuta regolarmente i nuovi tipi di nodi di cache disponibili e se ha senso, dal punto di vista dei costi e delle metriche operative, migrare il parco istanze a nuovi tipi di nodi di cache.

[Risorse]:

- [OBJECT IDLETIME](#)
- [Prezzi di Amazon ElastiCache](#)

Procedure e best practice comuni per la risoluzione dei problemi

Argomenti

- [Problemi di connessione](#)
- [Errori del client Redis](#)
- [Risoluzione dei problemi di latenza elevata in Serverless ElastiCache](#)
- [Risoluzione dei problemi di limitazione in Serverless ElastiCache](#)
- [Argomenti correlati](#)

Problemi di connessione

Se non riesci a connetterti alla ElastiCache cache, prendi in considerazione una delle seguenti opzioni:

1. Utilizzo di TLS: se si verifica un blocco della connessione durante il tentativo di connessione all' ElastiCache endpoint, è possibile che non si stia utilizzando TLS nel client. Se utilizzi ElastiCache Serverless, la crittografia in transito è sempre abilitata. Assicurati che il tuo client utilizzi TLS per connettersi alla cache. [Scopri di più sulla connessione a una cache abilitata per TLS qui.](#)

2. VPC: le ElastiCache cache sono accessibili solo dall'interno di un VPC. Assicurati che l'istanza EC2 da cui accedi alla cache e la ElastiCache cache siano create nello stesso VPC. In alternativa, devi abilitare il [peering VPC tra](#) il VPC in cui risiede l'istanza EC2 e il VPC in cui stai creando la cache.
3. Gruppi di sicurezza: ElastiCache utilizza i gruppi di sicurezza per controllare l'accesso alla cache. Considera i seguenti aspetti:
 - a. Assicurati che il gruppo di sicurezza utilizzato dalla ElastiCache cache consenta l'accesso in entrata dalla tua istanza EC2. Leggi [qui](#) per scoprire come configurare correttamente le regole in entrata nel tuo gruppo di sicurezza.
 - b. Assicurati che il gruppo di sicurezza utilizzato dalla ElastiCache cache consenta l'accesso alle porte della cache (6379 e 6380 per le versioni serverless e 6379 per impostazione predefinita per quelle progettate autonomamente). ElastiCache utilizza queste porte per accettare i comandi Redis. Scopri di più su come configurare l'accesso alle porte [qui](#).

Errori del client Redis

ElastiCache Serverless è accessibile solo tramite client Redis che supportano il protocollo Redis in modalità cluster. È possibile accedere ai cluster progettati autonomamente dai client Redis in entrambe le modalità, a seconda della configurazione del cluster.

Se riscontri errori Redis nel tuo client, considera quanto segue:

1. Modalità cluster: se si verificano errori CROSSLOT o errori con il comando [SELECT](#) Redis, è possibile che si stia tentando di accedere a una cache Cluster Mode Enabled con un client Redis che non supporta il protocollo Redis Cluster. ElastiCache Serverless supporta solo client Redis che supportano il protocollo cluster Redis. Se desideri utilizzare Redis in «Cluster Mode Disabled» (CMD), devi progettare il tuo cluster.
2. Errori CROSSLOT: se si verifica l'ERR CROSSLOT Keys in request don't hash to the same slot errore, è possibile che si stia tentando di accedere a chiavi che non appartengono allo stesso slot in una cache in modalità Cluster. Come promemoria, ElastiCache Serverless funziona sempre in modalità Cluster. Le operazioni a più chiavi, le transazioni o gli script Lua che coinvolgono più chiavi sono consentite solo se tutte le chiavi coinvolte si trovano nello stesso slot di hash.

[Per ulteriori best practice sulla configurazione dei client Redis, consulta questo post sul blog.](#)

Risoluzione dei problemi di latenza elevata in Serverless ElastiCache

Se il tuo carico di lavoro sembra presentare un'elevata latenza, puoi analizzare le `SuccessfulWriteRequestLatency` metriche CloudWatch `SuccessfulReadRequestLatency` e per verificare se la latenza è correlata a Serverless. ElastiCache Queste metriche misurano la latenza interna a ElastiCache Serverless: la latenza lato client e i tempi di viaggio di rete tra il client e l'endpoint Serverless non sono inclusi. ElastiCache

Alcune variabilità e i picchi occasionali non dovrebbero essere motivo di preoccupazione. Tuttavia, se la `Average` statistica mostra un forte aumento e persiste, dovresti controllare la `Personal Health Dashboard AWS Health Dashboard` e la tua `Personal Health Dashboard` per ulteriori informazioni. Se necessario, valuta la possibilità di aprire una richiesta di supporto con `AWS Support`

Prendi in considerazione le seguenti best practice e strategie per ridurre la latenza:

- **Abilita Read from Replica:** se la tua applicazione lo consente, ti consigliamo di abilitare la funzionalità «Read from Replica» nel tuo client Redis per scalare le letture e ottenere una latenza inferiore. Se abilitata, ElastiCache Serverless tenta di indirizzare le richieste di lettura verso i nodi di cache di replica che si trovano nella stessa zona di disponibilità (AZ) del client, evitando così la latenza di rete Cross-AZ. Tieni presente che l'attivazione della funzionalità Read from Replica nel client significa che l'applicazione accetta l'eventuale coerenza dei dati. L'applicazione potrebbe ricevere dati più vecchi per qualche tempo se si tenta di leggere dopo aver scritto su una chiave.
- **Assicurati che l'applicazione sia distribuita nelle stesse AZ della cache:** potresti osservare una latenza lato client più elevata se l'applicazione non è distribuita negli stessi AZ della cache. Quando si crea una cache serverless, è possibile fornire le sottoreti da cui l'applicazione accederà alla cache e ElastiCache Serverless crea endpoint VPC in tali sottoreti. Assicurati che l'applicazione sia distribuita nelle stesse AZ. In caso contrario, l'applicazione potrebbe subire un hop Cross-AZ quando accede alla cache, con conseguente maggiore latenza lato client.
- **Riutilizza le connessioni:** le richieste ElastiCache serverless vengono effettuate tramite una connessione TCP abilitata per TLS utilizzando il protocollo RESP. L'avvio della connessione (inclusa l'autenticazione della connessione, se configurata) richiede tempo, quindi la latenza della prima richiesta è superiore a quella tipica. Le richieste su una connessione già inizializzata offrono ElastiCache una latenza costantemente bassa. Per questo motivo, dovresti prendere in considerazione l'utilizzo del pool di connessioni o il riutilizzo delle connessioni Redis esistenti.
- **Velocità di scalabilità:** ElastiCache Serverless si ridimensiona automaticamente all'aumentare della frequenza delle richieste. Un aumento improvviso e significativo della frequenza di richieste, superiore alla velocità di scalabilità di ElastiCache Serverless, può comportare una latenza elevata

per qualche tempo. ElastiCache In genere, Serverless può aumentare rapidamente la frequenza di richieste supportata, impiegando fino a 10-12 minuti per raddoppiare la frequenza delle richieste.

- Ispeziona i comandi a esecuzione prolungata: alcuni comandi Redis, inclusi gli script Lua o i comandi su strutture di dati di grandi dimensioni, possono essere eseguiti a lungo. Per identificare questi comandi, ElastiCache pubblica metriche a livello di comando. Con [ElastiCache Serverless](#) puoi usare le metriche. BasedECPUs
- Richieste limitate: quando le richieste vengono limitate in ElastiCache Serverless, è possibile che si verifichi un aumento della latenza lato client nell'applicazione. [Quando le richieste vengono limitate in ElastiCache Serverless, dovresti notare un aumento della metrica Serverless. ThrottledRequests ElastiCache](#) Consulta la sezione seguente per la risoluzione dei problemi relativi alle richieste limitate.
- Distribuzione uniforme di chiavi e richieste: in ElastiCache Redis, una distribuzione non uniforme di chiavi o richieste per slot può comportare un hot slot che può comportare una latenza elevata. ElastiCache Serverless supporta fino a 30.000 ECPU al secondo (90.000 ECPU al secondo quando si utilizza Read from Replica) su un singolo slot, in un carico di lavoro che esegue semplici comandi SET/GET. Ti consigliamo di valutare la distribuzione delle chiavi e delle richieste tra gli slot e di garantire una distribuzione uniforme se la frequenza delle richieste supera questo limite.

Risoluzione dei problemi di limitazione in Serverless ElastiCache

Nelle architetture orientate ai servizi e nei sistemi distribuiti, la limitazione della velocità con cui le chiamate API vengono elaborate dai vari componenti del servizio si chiama limitazione (della larghezza di banda della rete). Ciò attenua i picchi, controlla le discrepanze nella velocità di trasmissione dei componenti e consente ripristini più prevedibili in caso di eventi operativi imprevisti. ElastiCache Serverless è progettato per questi tipi di architetture e la maggior parte dei client Redis dispone di nuovi tentativi integrati per le richieste limitate. Un certo grado di limitazione (della larghezza di banda della rete) non è necessariamente un problema per l'applicazione, ma una limitazione (della larghezza di banda della rete) persistente di una parte sensibile alla latenza del flusso di lavoro dei dati può influire negativamente sull'esperienza dell'utente e ridurre l'efficienza complessiva del sistema.

[Quando le richieste vengono limitate in Serverless, dovresti notare un aumento della ElastiCache metrica Serverless. ThrottledRequests ElastiCache](#) Se notate un numero elevato di richieste limitate, considerate quanto segue:

- **Velocità di scalabilità:** ElastiCache Serverless si ridimensiona automaticamente man mano che si acquisiscono più dati o si aumenta la frequenza delle richieste. Se l'applicazione si adatta più velocemente della scalabilità Serverless, le richieste potrebbero essere limitate mentre ElastiCache Serverless si ridimensiona per adattarsi al carico di lavoro. ElastiCache In genere, la modalità serverless consente di aumentare rapidamente le dimensioni di storage, impiegando fino a 10-12 minuti per raddoppiare le dimensioni di archiviazione nella cache.
- **Distribuzione uniforme di chiavi e richieste:** in ElastiCache Redis, una distribuzione non uniforme di chiavi o richieste per slot può causare un hot slot. Un hot slot può comportare una limitazione delle richieste se la frequenza delle richieste verso un singolo slot supera le 30.000 ECPU al secondo, in un carico di lavoro che esegue semplici comandi SET/GET.
- **Leggi dalla replica:** se l'applicazione lo consente, valuta la possibilità di utilizzare la funzione «Leggi dalla replica». La maggior parte dei client Redis può essere configurata per «scalare le letture» per indirizzare le letture ai nodi di replica. Questa funzionalità consente di scalare il traffico di lettura. Inoltre, ElastiCache Serverless indirizza automaticamente la lettura dalle richieste di replica ai nodi nella stessa zona di disponibilità dell'applicazione, con conseguente riduzione della latenza. Quando Read from Replica è abilitato, è possibile ottenere fino a 90.000 ECPU al secondo su un singolo slot, per carichi di lavoro con semplici comandi SET/GET.

Argomenti correlati

- [Ulteriori passaggi per la risoluzione dei problemi](#)
- [the section called “Best practice e strategie di caching”](#)

Ulteriori passaggi per la risoluzione dei problemi

I seguenti elementi devono essere verificati durante la risoluzione dei problemi di connettività persistenti con ElastiCache:

Argomenti

- [Gruppi di sicurezza](#)
- [Liste di controllo accessi \(ACL\) di rete](#)
- [Tabelle di instradamento](#)
- [Risoluzione DNS](#)
- [Identificazione dei problemi con la diagnostica lato server](#)

- [Convalida della connettività di rete](#)
- [Limiti relativi alla rete](#)
- [Utilizzo CPU](#)
- [Connessioni terminate dal lato server](#)
- [Risoluzione dei problemi lato client per istanze Amazon EC2](#)
- [Dissezione del tempo necessario per completare una singola richiesta](#)

Gruppi di sicurezza

I gruppi di sicurezza sono firewall virtuali che proteggono il ElastiCache client (istanza EC2, AWS Lambda funzione, contenitore Amazon ECS, ecc.) e la cache. ElastiCache I gruppi di sicurezza sono dotati di stato, il che significa che dopo che il traffico in entrata o in uscita è consentito, le risposte per tale traffico verranno automaticamente autorizzate nel contesto di tale gruppo di sicurezza specifico.

La funzionalità con stato richiede che il gruppo di sicurezza monitori tutte le connessioni autorizzate ed esiste un limite per le connessioni monitorate. Se il limite viene raggiunto, le nuove connessioni avranno esito negativo. Consulta la sezione dedicata alla risoluzione dei problemi per sapere se i limiti sono stati raggiunti dal client o dal lato. ElastiCache

È possibile assegnare contemporaneamente un singolo gruppo di sicurezza al client e al ElastiCache cluster o singoli gruppi di sicurezza per ciascuno.

In entrambi i casi, è necessario consentire il traffico TCP in uscita sulla ElastiCache porta dall'origine e il traffico in entrata sulla stessa porta verso. ElastiCache La porta predefinita è 11211 per Memcached e 6379 per Redis. Per impostazione predefinita, i gruppi di sicurezza autorizzano tutto il traffico in uscita. In questo caso, è necessaria solo la regola in entrata nel gruppo di sicurezza di destinazione.

Per ulteriori informazioni, consulta [Modelli di accesso per l'accesso a un ElastiCache cluster in un Amazon VPC](#).

Liste di controllo accessi (ACL) di rete

Liste di controllo accessi di rete (ACL) sono regole senza stato. Il traffico deve essere consentito in entrambe le direzioni (in entrata e in uscita) per avere esito positivo. Le ACL di rete vengono assegnate alle sottoreti, non a risorse specifiche. È possibile assegnare lo stesso ACL ElastiCache e la stessa risorsa client, specialmente se si trovano nella stessa sottorete.

Per impostazione predefinita, le ACL di rete consentono tutti i traffici. Tuttavia, è possibile personalizzarle per negare o consentire il traffico. Inoltre, la valutazione delle regole delle ACL è sequenziale, ossia la regola con il numero più basso corrispondente al traffico consentirà o negherà tale traffico. La configurazione minima per consentire il traffico Redis è:

ACL di rete lato client:

- Regole in entrata:
 - Numero della regola: preferibilmente inferiore a qualsiasi regola di diniego;
 - Tipo: Regola TCP personalizzata;
 - Protocollo: TCP
 - Intervallo porte: 1024-65535
 - Fonte: 0.0.0.0/0 (o crea regole individuali per le sottoreti del cluster) ElastiCache
 - Consenti/Nega: Consenti
-
- Regole in uscita:
 - Numero della regola: preferibilmente inferiore a qualsiasi regola di diniego;
 - Tipo: Regola TCP personalizzata;
 - Protocollo: TCP
 - Intervallo porte: 6379
 - Fonte: 0.0.0.0/0 (o le sottoreti del cluster. ElastiCache Tieni presente che l'utilizzo di IP specifici può creare problemi in caso di failover o scalabilità del cluster)
 - Consenti/Nega: Consenti

ElastiCache ACL di rete:

- Regole in entrata:
- Numero della regola: preferibilmente inferiore a qualsiasi regola di diniego;
- Tipo: Regola TCP personalizzata;
- Protocollo: TCP
- Intervallo porte: 6379
- Fonte: 0.0.0.0/0 (o crea regole individuali per le sottoreti del cluster) ElastiCache
- Consenti/Nega: Consenti

- Regole in uscita:
- Numero della regola: preferibilmente inferiore a qualsiasi regola di diniego;
- Tipo: Regola TCP personalizzata;
- Protocollo: TCP
- Intervallo porte: 1024-65535
- Fonte: 0.0.0.0/0 (o le sottoreti del cluster. ElastiCache Tieni presente che l'utilizzo di IP specifici può creare problemi in caso di failover o scalabilità del cluster)
- Consenti/Nega: Consenti

Per ulteriori informazioni, consultare [ACL di rete](#).

Tabelle di instradamento

Analogamente alle ACL di rete, ogni sottorete può avere tabelle di routing diverse. Se i client e il ElastiCache cluster si trovano in sottoreti diverse, assicurati che le tabelle di routing consentano loro di raggiungersi.

Gli ambienti più complessi, che coinvolgono più routing dinamici di VPC, o firewall di rete, possono risultare difficili da risolvere. Consultare [Convalida della connettività di rete](#) per verificare che le impostazioni di rete siano appropriate.

Risoluzione DNS

ElastiCache fornisce gli endpoint del servizio in base ai nomi DNS. Gli endpoint disponibili sono Configuration, Primary, Reader eNode. Per maggiori informazioni, consultare [Trovare gli endpoint di connessione](#).

In caso di failover o di modifica del cluster, l'indirizzo associato al nome dell'endpoint potrebbe cambiare e verrà aggiornato automaticamente.

Le impostazioni DNS personalizzate (ad esempio, che non utilizzano il servizio DNS VPC) potrebbero non essere a conoscenza dei ElastiCache nomi DNS forniti. Assicurati che il tuo sistema sia in grado di risolvere correttamente gli ElastiCache endpoint utilizzando strumenti di sistema come dig (come mostrato di seguito) o nslookup

```
$ dig +short example.xxxxxx.ng.0001.use1.cache.amazonaws.com
example-001.xxxxxx.0001.use1.cache.amazonaws.com.
```

```
1.2.3.4
```

È inoltre possibile forzare la risoluzione dei nomi tramite il servizio del DNS VPC:

```
$ dig +short example.xxxxxx.ng.0001.use1.cache.amazonaws.com @169.254.169.253
example-001.tihewd.0001.use1.cache.amazonaws.com.
1.2.3.4
```

Identificazione dei problemi con la diagnostica lato server

CloudWatch le metriche e le informazioni sul tempo di funzionamento del ElastiCache motore sono fonti o informazioni comuni per identificare potenziali fonti di problemi di connessione. Una buona analisi inizia comunemente con i seguenti elementi:

- **Utilizzo della CPU:** Redis è un'applicazione multithread. Tuttavia, l'esecuzione di ogni comando avviene in un singolo thread (principale). Per questo motivo, ElastiCache fornisce le metriche `CPUUtilization` e `EngineCPUUtilization`. `EngineCPUUtilization` fornisce l'utilizzo della CPU dedicato al processo Redis e l'utilizzo su tutte `CPUUtilization` le vCPU. I nodi con più di una vCPU di solito hanno valori diversi per `CPUUtilization` e `EngineCPUUtilization`, il secondo è comunemente più alto. Un elevato `EngineCPUUtilization` può essere causato da un numero elevato di richieste o operazioni complesse che richiedono una notevole quantità di tempo di CPU per il completamento. È possibile identificare entrambi con i seguenti elementi:
 - **Numero elevato di richieste:** verifica la presenza di aumenti su altri parametri che corrispondono al modello `EngineCPUUtilization`. I parametri utili sono:
 - **CacheHits e CacheMisses:** il numero di richieste corrette o richieste che non hanno trovato un elemento valido nella cache. Se il rapporto tra perdite e successi è elevato, l'applicazione sta sprestando tempo e risorse con richieste infruttuose.
 - **SetTypeCmds e GetTypeCmds:** questi parametri correlati con `EngineCPUUtilization` possono aiutare a capire se il carico è significativamente più alto per le richieste di scrittura, misurato da `SetTypeCmds`, o letture, misurate da `GetTypeCmds`. Se il carico è prevalentemente di lettura, l'utilizzo di più repliche di lettura può bilanciare le richieste su più nodi e risparmiare il primario per le scritture. Nei cluster disattivati in modalità cluster, è possibile utilizzare le repliche di lettura creando una configurazione di connessione aggiuntiva nell'applicazione utilizzando l'endpoint di lettura. ElastiCache Per maggiori informazioni, consultare [Trovare gli endpoint di connessione](#). Le operazioni di lettura devono essere inviate a questa connessione aggiuntiva. Le operazioni di scrittura verranno eseguite attraverso l'endpoint primario regolare. In modalità cluster abilitata, è consigliabile utilizzare una libreria

che supporti le repliche di lettura in modalità nativa. Con i flag giusti, la libreria sarà in grado di individuare automaticamente la topologia del cluster, i nodi di replica, abilitare le operazioni di lettura tramite il comando di Redis [READONLY](#) e inviare le richieste di lettura alle repliche.

- Elevato numero di connessioni:
- `CurrConnections` e `NewConnections`: `CurrConnection` è il numero di connessioni stabilite al momento della raccolta dei datapoint, mentre `NewConnections` mostra quante connessioni sono state create nel periodo.

La creazione e la gestione delle connessioni implica un notevole overhead della CPU. Inoltre, l'handshake a tre vie TCP richiesto per creare nuove connessioni influirà negativamente sui tempi di risposta complessivi.

Un ElastiCache nodo con migliaia di dati `NewConnections` al minuto indica che una connessione viene creata e utilizzata con pochi comandi, il che non è ottimale. Mantenere le connessioni stabilite e riutilizzarle per nuove operazioni è una best practice. Ciò è possibile quando l'applicazione client supporta e implementa correttamente il pool di connessioni o le connessioni persistenti. Con il pool di connessioni, il numero di `currConnections` non ha grandi variazioni, e le `NewConnections` dovrebbero essere il più basso possibile. Redis fornisce prestazioni ottimali con un numero ridotto di `currConnections`. Mantenere `currConnection` nell'ordine di decine o centinaia riduce al minimo l'utilizzo di risorse per supportare singole connessioni come buffer client e cicli CPU per servire la connessione.

- Velocità effettiva di rete
- Determina la larghezza di banda: ElastiCache i nodi hanno una larghezza di banda di rete proporzionale alla dimensione del nodo. Poiché le applicazioni hanno caratteristiche diverse, i risultati possono variare in base al carico di lavoro. Ad esempio, le applicazioni con un elevato tasso di richieste di piccole dimensioni tendono a influire di più sull'utilizzo della CPU rispetto alla velocità effettiva di rete, mentre le chiavi più grandi causeranno un maggiore utilizzo della rete. Per questo motivo, è consigliabile testare i nodi con il carico di lavoro effettivo per una migliore comprensione dei limiti.

La simulazione del carico dall'applicazione fornirebbe risultati più accurati. Tuttavia, gli strumenti relativi al valore di riferimento possono dare una buona idea dei limiti.

- Nei casi in cui le richieste sono prevalentemente letture, l'utilizzo delle repliche per le operazioni di lettura riduce il carico sul nodo primario. Se il caso d'uso è prevalentemente scritture, l'uso di molte repliche amplificherà l'utilizzo della rete. Per ogni byte scritto nel nodo primario, N byte verranno inviati alle repliche, essendo N il numero di repliche. La best practice

per carichi di lavoro ad alta intensità di scrittura è quella di utilizzare Redis con ElastiCache la modalità cluster abilitata, in modo che le scritture possano essere bilanciate su più shard o scalare fino a un tipo di nodo con maggiori funzionalità di rete.

- I CloudWatchmetrics `NetworkBytesIn` e `NetworkBytesOut` forniscono rispettivamente la quantità di dati in entrata o in uscita dal nodo. `ReplicationBytes` è il traffico dedicato alla replica dei dati.

Per ulteriori informazioni, consulta [Limiti relativi alla rete](#).

- Comandi complessi: i comandi di Redis sono serviti su un singolo thread, ossia le richieste vengono servite in sequenza. Un singolo comando lento può influire su altre richieste e connessioni, culminando in time-out. L'uso di comandi che agiscono su più valori, chiavi o tipi di dati deve essere fatto con attenzione. Le connessioni possono essere bloccate o terminate a seconda del numero di parametri o delle dimensioni dei valori di input o output.

Un esempio famigerato è il comando `KEYS`. Esso spazza l'intero keyspace alla ricerca di un dato modello e blocca l'esecuzione di altri comandi durante la sua esecuzione. Redis utilizza la notazione "Big O" per descrivere la complessità dei suoi comandi.

Il comando delle chiavi ha complessità di tempo $O(N)$, dove N è il numero di chiavi presenti nel database. Pertanto, maggiore è il numero di chiavi, più lento sarà il comando. `KEYS` può causare problemi in diversi modi: se non viene utilizzato alcun modello di ricerca, il comando restituirà tutti i nomi delle chiavi disponibili. Nei database con migliaia o milioni di elementi, verrà creato un output enorme e inonderà i buffer di rete.

Se viene utilizzato un modello di ricerca, solo le chiavi corrispondenti al modello torneranno al client. Tuttavia, il motore continua a spazzare l'intero keyspace alla sua ricerca e il tempo necessario per completare il comando sarà lo stesso.

Un'alternativa per `KEYS` è il comando `SCAN`. Si ripete sul keyspace e limita le iterazioni in un numero specifico di elementi, evitando blocchi prolungati sul motore.

La scansione ha il parametro `COUNT`, utilizzato per impostare la dimensione dei blocchi di iterazione. Il valore di default è 10 (10 elementi per iterazione).

A seconda del numero di elementi nel database, piccoli blocchi di valori `COUNT` richiedono più iterazioni per completare una scansione completa e valori più grandi manterranno il motore occupato più a lungo ad ogni iterazione. Mentre piccoli valori di conteggio renderanno `SCAN` più

lento su database di grandi dimensioni, valori più grandi possono causare gli stessi problemi menzionati per KEYS.

Ad esempio, l'esecuzione del comando SCAN con valore di conteggio di 10 richiederà 100.000 ripetizioni su un database con 1 milione di chiavi. Se il Round Trip Time medio della rete è di 0,5 millisecondi, verranno spesi circa 50.000 millisecondi (50 secondi) per trasferire le richieste.

D'altra parte, se il valore di conteggio fosse 100.000, sarebbe necessaria una singola iterazione e solo 0,5 ms sarebbero spesi per trasferirla. Tuttavia, il motore sarebbe completamente bloccato per altre operazioni fino a quando il comando non termina di spazzare tutto il keypace.

Oltre a KEYS, molti altri comandi sono potenzialmente dannosi se non utilizzati correttamente. Per visualizzare un elenco di tutti i comandi e la rispettiva complessità temporale, andare all'indirizzo <https://redis.io/commands>.

Esempi di potenziali problemi:

- **Script Lua:** Redis fornisce un interprete Lua incorporato, consentendo l'esecuzione di script sul lato server. Gli script Lua su Redis vengono eseguiti a livello di motore e sono atomici per definizione, ossia nessun altro comando o script avrà il permesso di eseguire mentre uno script è in esecuzione. Gli script Lua offrono la possibilità di eseguire più comandi, algoritmi decisionali, analisi dei dati e altri direttamente sul motore di Redis. Mentre l'atomicità degli script e la possibilità di scaricare l'applicazione sono allettanti, gli script devono essere utilizzati con cura e per piccole operazioni. SÌ ElastiCache, il tempo di esecuzione degli script Lua è limitato a 5 secondi. Gli script che non sono stati scritti nel keypacei verranno terminati automaticamente dopo un periodo di 5 secondi. Per evitare il danneggiamento dei dati e le incongruenze, il nodo eseguirà il failover se l'esecuzione dello script non è stata completata in 5 secondi e durante l'esecuzione non è stata eseguita alcuna scrittura. Le [Transazioni](#) sono l'alternativa per garantire la coerenza di più modifiche chiave correlate in Redis. Una transazione consente l'esecuzione di un blocco di comandi, osservando le chiavi esistenti per le modifiche. Se una delle chiavi osservate cambia prima del completamento della transazione, tutte le modifiche vengono scartate.
- **Eliminazione di massa di elementi:** il comando DEL accetta più parametri, che sono i nomi di chiave da eliminare. Le operazioni di eliminazione sono sincrone e richiedono un tempo significativo della CPU se l'elenco dei parametri è grande, o contiene un elenco grande, un set ordinato o un hash (strutture di dati che contengono più sottovoci). In altre parole, anche la cancellazione di una singola chiave può richiedere tempo significativo se ha molti elementi. L'alternativa a DEL è UNLINK, che è un comando asincrono disponibile da Redis 4. UNLINK

deve essere preferito a DEL ogniqualvolta è possibile. A ElastiCache partire da Redis 6x, il `lazyfree-lazy-user-del` parametro fa sì che il DEL comando si comporti come quando è abilitato. UNLINK Per ulteriori informazioni, consultare la sezione [Modifica dei parametri di Redis 6.0](#).

- Comandi che agiscono su più chiavi: DEL è stato menzionato prima come comando che accetta più argomenti e il suo tempo di esecuzione sarà direttamente proporzionale a quello. Tuttavia, Redis fornisce molti altri comandi che funzionano in modo simile. Ad esempio, MSET e MGET consentono l'inserimento o il recupero di più chiavi di stringa contemporaneamente. Il loro utilizzo può essere utile per ridurre la latenza di rete intrinseca a più comandi SET o GET individuali. Tuttavia, un ampio elenco di parametri influenzerà l'utilizzo della CPU.

Sebbene l'utilizzo della CPU da solo non sia la causa di problemi di connettività, dedicare troppo tempo all'elaborazione di uno o pochi comandi su più chiavi può causare l'utilizzo fuori limite di altre richieste e aumentare l'utilizzo complessivo della CPU.

Il numero di chiavi e le loro dimensioni influenzeranno la complessità del comando e, di conseguenza, il tempo di completamento.

Altri esempi di comandi che possono agire su più chiavi: HMGET, HMSET, MSETNX, PFCOUNT, PFMERGE, SDIFF, SDIFFSTORE, SINTER, SINTERSTORE, SUNION, SUNIONSTORE, TOUCH, ZDIFF, ZDIFFSTORE, ZINTER o ZINTERSTORE.

- Comandi che agiscono su più tipi di dati: Redis fornisce anche comandi che agiscono su una o più chiavi, indipendentemente dal tipo di dati. ElastiCache for Redis fornisce la metrica `KeyBasedCmds` per monitorare tali comandi. Questo parametro somma l'esecuzione dei seguenti comandi nel periodo selezionato:
 - Complessità $O(N)$:
 - KEYS
 - $O(1)$
 - EXISTS
 - OBJECT
 - PTTL
 - RANDOMKEY
 - TTL
 - TYPE
 - EXPIRE

- EXPIREAT
- MOVE
- PERSIST
- PEXPIRE
- PEXPIREAT
- UNLINK ($O(N)$) per recuperare memoria. Tuttavia il processo di recupero della memoria avviene in un thread separato e non blocca il motore
- Tempi di complessità differenti a seconda del tipo di dati:
 - DEL
 - DUMP
 - RENAME è considerato un comando con complessità $O(1)$, ma esegue DEL internamente. Il tempo di esecuzione varia a seconda delle dimensioni della chiave rinominata.
 - RENAMENX
 - RESTORE
 - SORT
- Big hash: Hash è un tipo di dati che consente una singola chiave con più sottoelementi chiave-valore. Ogni hash può archiviare 4.294.967.295 elementi e le operazioni su hash di grandi dimensioni possono diventare costose. Analogamente a KEYS, gli hash hanno il comando HKEYS con complessità temporale $O(N)$, dove N è il numero di elementi nell'hash. HSCAN deve essere preferito a HKEYS per evitare comandi di lunga durata. HDEL, HGETALL, HMGET, HMSET e HVALS sono comandi che dovrebbero essere usati con cautela su hash di grandi dimensioni.
- Altre strutture Big Data: oltre agli hash, altre strutture di dati possono richiedere un uso intensivo della CPU. Insiemi, elenchi, insiemi ordinati e hyperloglog possono anche richiedere molto tempo per essere manipolati a seconda delle dimensioni e dei comandi utilizzati. Per ulteriori informazioni su tali comandi, consultare <https://redis.io/commands>.

Convalida della connettività di rete

Dopo aver esaminato le configurazioni di rete relative alla risoluzione del DNS, ai gruppi di sicurezza, alle ACL di rete e alle tabelle di routing, la connettività può essere convalidata con VPC Reachability Analyzer e gli strumenti di sistema.

Reachability Analyzer verificherà la connettività di rete e confermerà se tutti i requisiti e le autorizzazioni sono soddisfatti. Per i test seguenti è necessario l'ENI ID (Elastic Network Interface Identification) di uno dei ElastiCache nodi disponibili nel tuo VPC. È possibile trovarlo eseguendo le seguenti operazioni:

1. Andare su <https://console.aws.amazon.com/ec2/v2/home?#NIC:>
2. Filtra l'elenco delle interfacce in base al nome del ElastiCache cluster o all'indirizzo IP ottenuto dalle convalide DNS precedenti.
3. Annotare o comunque salvare l'ID ENI. Se vengono visualizzate più interfacce, rivedi la descrizione per confermare che appartengano al ElastiCache cluster corretto e scegline una.
4. Passare alla fase successiva.
5. Create un percorso di analisi su [https://console.aws.amazon.com/vpc/home? # ReachabilityAnalyzer](https://console.aws.amazon.com/vpc/home?#ReachabilityAnalyzer) e scegliete le seguenti opzioni:
 - Tipo di fonte: scegli l'istanza se il tuo ElastiCache client funziona su un'istanza Amazon EC2 o un'interfaccia di rete se utilizza un altro servizio, come AWS Fargate Amazon ECS con rete awsvpc AWS Lambda, ecc.) e il rispettivo ID di risorsa (istanza EC2 o ID ENI);
 - Tipo di destinazione: Scegliere Interfaccia di rete e selezionare il ENI di ElastiCache dall'elenco.
 - Porta di destinazione: specifica 6379 per Redis o 11211 ElastiCache per Memcached. ElastiCache Queste sono le porte definite con la configurazione di default e questo esempio presuppone che non vengano modificate.
 - Protocollo: TCP

Creare il percorso di analisi e attendere alcuni minuti per il risultato. Se lo stato non è raggiungibile, aprire i dettagli dell'analisi ed esaminare Esplora analisi per i dettagli in cui le richieste sono state bloccate.

Se i test di raggiungibilità sono stati superati, procedere alla verifica a livello di sistema operativo.

Per convalidare la connettività TCP sulla porta di ElastiCache servizio: su Amazon Linux, Nping è disponibile nel pacchetto nmap e può testare la connettività TCP sulla ElastiCache porta, oltre a fornire il tempo di andata e ritorno della rete per stabilire la connessione. Usalo per convalidare la connettività di rete e la latenza corrente verso il cluster, come illustrato di seguito: ElastiCache

```
$ sudo nping --tcp -p 6379 example.xxxxxx.ng.0001.use1.cache.amazonaws.com
```

```
Starting Nping 0.6.40 ( http://nmap.org/nping ) at 2020-12-30 16:48 UTC
SENT (0.0495s) TCP ...
(Output suppressed )

Max rtt: 0.937ms | Min rtt: 0.318ms | Avg rtt: 0.449ms
Raw packets sent: 5 (200B) | Rcvd: 5 (220B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 4.08 seconds
```

Per impostazione predefinita, nping invia 5 sonde con un ritardo di 1 secondo tra di loro. È possibile utilizzare l'opzione “-c” per aumentare il numero di sonde e “--delay” per cambiare il tempo di invio di un nuovo test.

Se i test con nping falliscono e i test del VPC Reachability Analyzer passano, chiedere all'amministratore di sistema di esaminare le possibili regole firewall basate su host, regole di routing asimmetriche o qualsiasi altra possibile restrizione a livello di sistema operativo.

Sulla ElastiCache console, controlla se Encryption in-transit è abilitata nei dettagli del cluster ElastiCache . Se la crittografia in transito è abilitata, verificare se la sessione della TLS può essere stabilita con il seguente comando:

```
openssl s_client -connect example.xxxxxx.use1.cache.amazonaws.com:6379
```

Un output esteso è previsto se la connessione e la negoziazione relativa alla TLS hanno esito positivo. Controllare il codice restituito disponibile nell'ultima riga, il valore deve essere 0 (ok). Se openssl restituisce qualcosa di diverso, controllare il motivo dell'errore su <https://www.openssl.org/docs/man1.0.2/man1/verify.html#DIAGNOSTICS>.

Se tutti i test dell'infrastruttura e del sistema operativo sono stati superati ma l'applicazione non riesce ancora a connettersi ElastiCache, verifica se le configurazioni dell'applicazione sono conformi alle impostazioni. ElastiCache Gli errori più comuni sono:

- L'applicazione non supporta la modalità ElastiCache cluster e la modalità cluster ElastiCache è abilitata;
- L'applicazione non supporta TLS/SSL e ElastiCache ha la crittografia in transito abilitata;
- L'applicazione supporta TLS/SSL ma non dispone dei flag di configurazione corretti o delle certification authority attendibili;

Limiti relativi alla rete

- Numero massimo di connessioni: ci sono limiti rigidi per le connessioni simultanee. Ogni ElastiCache nodo consente fino a 65.000 connessioni simultanee su tutti i client. Questo limite può essere monitorato tramite le `CurrentConnections` metriche su CloudWatch. Tuttavia, i client hanno anche i loro limiti per le connessioni in uscita. Su Linux, controllare l'intervallo di porte effimere consentito con il comando:

```
# sysctl net.ipv4.ip_local_port_range
net.ipv4.ip_local_port_range = 32768 60999
```

Nell'esempio precedente, saranno consentite 28231 connessioni dalla stessa origine, allo stesso IP (ElastiCache nodo) e porta di destinazione. Il comando seguente mostra quante connessioni esistono per un ElastiCache nodo specifico (IP 1.2.3.4):

```
ss --numeric --tcp state connected "dst 1.2.3.4 and dport == 6379" | grep -vE
'^State' | wc -l
```

Se il numero è troppo alto, il sistema potrebbe sovraccaricarsi tentando di elaborare le richieste di connessione. Si consiglia di prendere in considerazione l'implementazione di tecniche come il pool di connessioni o connessioni persistenti per gestire meglio le connessioni. Ogniqualvolta è possibile, configurare il pool di connessioni per limitare il numero massimo di connessioni ad alcune centinaia. Inoltre, la logica di back-off per gestire i timeout o altre eccezioni relative alla connessione sono consigliabili per evitare interruzioni di connessione in caso di problemi.

- Limiti del traffico di rete: controlla le seguenti [CloudWatch metriche per Redis per](#) identificare i possibili limiti di rete raggiunti sul nodo: ElastiCache
 - `NetworkBandwidthInAllowanceExceeded` / `NetworkBandwidthOutAllowanceExceeded`: pacchetti di rete sagomati perché la velocità effettiva ha superato il limite di larghezza di banda aggregata.

È importante notare che ogni byte scritto nel nodo primario verrà replicato in N repliche, dove N è il numero di repliche. I cluster con tipi di nodi piccoli, repliche multiple e richieste di scrittura ad alta intensità potrebbero non essere in grado di gestire il backlog di replica. In questi casi, è best practice dimensionare verso l'alto (cambiare il tipo di nodo), aumentare orizzontalmente (aggiungere partizioni in cluster in modalità cluster abilitata), ridurre il numero di repliche o ridurre al minimo il numero di scritture.

- `NetworkConntrackAllowanceExceeded`: pacchetti sagomati perché è stato superato il numero massimo di connessioni monitorate in tutti i gruppi di sicurezza assegnati al nodo. Le nuove connessioni probabilmente non avranno esito positivo durante questo periodo.
- `NetworkPackets PerSecondAllowanceExceeded`: Numero massimo di pacchetti al secondo superato. I carichi di lavoro basati su un elevato tasso di richieste molto piccole possono raggiungere questo limite prima della larghezza di banda massima.

I parametri sopra riportati sono il modo ideale per confermare che i nodi raggiungono i loro limiti di rete. Tuttavia, i limiti sono identificabili anche dai plateau sui parametri di rete.

Se i plateau vengono osservati per periodi prolungati, saranno probabilmente seguiti da ritardo di replica, aumento dei byte utilizzati per la cache, calo della memoria liberabile, elevato swap e utilizzo della CPU. Le istanze Amazon EC2 hanno anche limiti di rete che possono essere monitorati attraverso [Parametri di driver ENA](#). Le istanze Linux con supporto di rete avanzato e driver ENA 2.2.10 o versioni successive possono esaminare i contatori del limite con il comando:

```
# ethtool -S eth0 | grep "allowance_exceeded"
```

Utilizzo CPU

La metrica di utilizzo della CPU è il punto di partenza dell'indagine e i seguenti elementi possono aiutare a restringere i possibili problemi secondari: ElastiCache

- **Redis SlowLogs**: la configurazione ElastiCache predefinita mantiene gli ultimi 128 comandi che hanno richiesto più di 10 millisecondi per essere completati. La cronologia dei comandi lenti viene mantenuta durante il tempo di esecuzione del motore e andrà persa in caso di guasto o riavvio. Se l'elenco raggiunge 128 voci, i vecchi eventi verranno rimossi per fare spazio a quelli nuovi. La dimensione dell'elenco degli eventi lenti e il tempo di esecuzione considerato lento può essere modificato tramite i parametri `slowlog-max-len` e `slowlog-log-slower-than` in un [gruppo di parametri personalizzati](#). L'elenco dei slow logs può essere recuperato eseguendo `SLOWLOG GET 128` sul motore, in cui 128 sono gli ultimi 128 comandi lenti segnalati. Ogni voce ha i campi riportati di seguito:

```
1) 1) (integer) 1 -----> Sequential ID
   2) (integer) 1609010767 --> Timestamp (Unix epoch time)of the Event
   3) (integer) 4823378 -----> Time in microseconds to complete the command.
   4) 1) "keys" -----> Command
```

```
2) "*" -----> Arguments
5) "1.2.3.4:57004"-> Source
```

L'evento di cui sopra è avvenuto il 26 dicembre, alle 19:26:07 UTC, ha impiegato 4,8 secondi (4,823 ms) per essere completato ed è stato causato dal comando KEYS richiesto dal client 1.2.3.4.

Su Linux, la marcatura oraria può essere convertita con la data del comando:

```
$ date --date='@1609010767'
Sat Dec 26 19:26:07 UTC 2020
```

Con Python:

```
>>> from datetime import datetime
>>> datetime.fromtimestamp(1609010767)
datetime.datetime(2020, 12, 26, 19, 26, 7)
```

Oppure su Windows con: PowerShell

```
PS D:\Users\user> [datetimeoffset]::FromUnixTimeSeconds('1609010767')
DateTime           : 12/26/2020 7:26:07 PM
UtcDateTime        : 12/26/2020 7:26:07 PM
LocalDateTime      : 12/26/2020 2:26:07 PM
Date               : 12/26/2020 12:00:00 AM
Day               : 26
DayOfWeek          : Saturday
DayOfYear          : 361
Hour               : 19
Millisecond        : 0
Minute            : 26
Month              : 12
Offset             : 00:00:00Ticks           : 637446075670000000
UtcTicks           : 637446075670000000
TimeOfDay          : 19:26:07
Year              : 2020
```

Molti comandi lenti in un breve periodo di tempo (stesso minuto o meno) sono motivo di preoccupazione. Esaminare la natura dei comandi e come possono essere ottimizzati (vedere esempi precedenti). Se i comandi con complessità temporale $O(1)$ vengono segnalati frequentemente, controllare gli altri fattori relativamente a un utilizzo elevato della CPU come menzionato in precedenza.

- **Metriche di latenza:** ElastiCache for Redis fornisce CloudWatch metriche per monitorare la latenza media per diverse classi di comandi. Il datapoint viene calcolato dividendo il numero totale di esecuzioni di comandi nella categoria per il tempo totale di esecuzione nel periodo. È importante comprendere che i risultati dei parametri di latenza sono un aggregato di più comandi. Un singolo comando può causare risultati imprevisti, come i timeout, senza un impatto significativo sui parametri. In questi casi, gli eventi di slow log sarebbero una fonte di informazioni più accurata. L'elenco seguente contiene i parametri di latenza disponibili e i rispettivi comandi che le riguardano.
 - **EvalBasedCmdsLatency:** relativo ai comandi Lua Script,;, `eval` `evalsha`
 - **GeoSpatialBasedCmdsLatency:** `geodist`, `geohash`, `geopos`, `georadius`, `georadiusbymember`, `geoadd`;
 - **GetTypeCmdsLatency:** comandi di lettura, indipendentemente dal tipo di dati;
 - **HashBasedCmdsLatency:** `hexists`, `hget`, `hgetall`, `hkeys`, `hlen`, `hmget`, `hvals`, `hstrlen`, `hdel`, `hincrby`, `hincrbyfloat`, `hset`, `hsetnx`;
 - **HyperLogLogBasedCmdsLatency:** `pfselftest`, `pfcount`, `pfdebug`, `pfadd`, `pfmerge`;
 - **KeyBasedCmdsLatency:** Comandi che possono agire su diversi tipi di dati: `dump` `existskeys`, `object`, `pttl`, `randomkey`, `ttdel`, `ttl`, `type`, `del`, `expire`, `expireat`, `move`, `persist`, `pexpire`
 - **ListBasedCmdsLatency:** `lindex`, `len`, `range`, `blpop`, `brpop`, `brpoplpush`, `linsert`, `lpop`, `push`, `lpushx`, `lrem`, `lset`, `ltrim`, `rpop`, `rpoplpush`, `rpush`, `rpushx`;
 - **PubSubBasedCmdsLatency:** `psubscribe`, `publish`, `pubsub`, annulla sottoscrizione, sottoscrizione, annullamento dell'iscrizione;
 - **SetBasedCmdsLatency:** `scard`, `sdiff`, `sinter`, `sismember`, `smembers`, `srandmember`, `sunion`, `sadd`, `sdiffstore`, `sinterstore`, `smove`, `spop`, `srem`, `sunionstore`;
 - **SetTypeCmdsLatency:** scrivere comandi, indipendentemente dal tipo di dati;
 - **SortedSetBasedCmdsLatency:** `zcard`, `zcount`, `zrange`, `zrangebyscore`, `zrank`, `zrevrange`, `zrevrangebyscore`, `zrevrank`, `zscore`, `zrangebylex`, `zrevrangebylex`, `zlexcount`, `zadd`, `zincrby`, `zinterstore`, `zrem`, `zremrangebyrank`, `zremrangebyscore`, `zunionstore`, `zremrangebylex`, `zpopmax`, `zpopmin`, `bzpopmin`, `bzpopmax`;

- **StringBasedCmdsLatency:** `bitcount`, `get`, `getbit`, `getrange`, `mget`, `strlen`, `substr`, `bitpos`, `append`, `bitop`, `bitfield`, `decr`, `decrby`, `getset`, `incr`, `incrby`, `incrbyfloat`, `mset`, `msetnx`, `psetex`, `set`, `setbit`, `setex`, `setnx`, `setrange`;
- **StreamBasedCmdsLatency:** `xrange`, `xrevrange`, `xlen`, `xread`, `xpending`, `xinfo`, `xadd`, `xgroup`, `readgroup`, `xack`, `xclaim`, `xdel`, `xtrim`, `xsetid`;
- **Comandi relativi al tempo di esecuzione di Redis:**
 - `info commandstats`: fornisce un elenco di comandi eseguiti dall'avvio del motore Redis, il numero di esecuzioni cumulative, il tempo totale di esecuzione e il tempo medio di esecuzione per comando;
 - `client list`: fornisce un elenco di client attualmente connessi e informazioni rilevanti come l'utilizzo dei buffer, l'ultimo comando eseguito, ecc.;
- **Backup e replica:** ElastiCache per le versioni Redis precedenti alla 2.8.22, utilizza un processo biforcuto per creare backup ed elaborare sincronizzazioni complete con le repliche. Questo metodo può incorrere in un sovraccarico di memoria significativo per casi d'uso intensivi di scrittura.

A partire da ElastiCache Redis 2.8.22, è stato introdotto un metodo di backup e replica senza `forkless`. AWS Il nuovo metodo potrebbe ritardare le scritture al fine di evitare errori. Entrambi i metodi possono causare periodi di maggiore utilizzo della CPU, portare a tempi di risposta più elevati e di conseguenza portare a timeout client durante la loro esecuzione. Controllare sempre se gli errori del client si verificano durante la finestra di backup o il parametro `SaveInProgress` è 1 nel periodo. Si consiglia di pianificare la finestra di backup per periodi di basso utilizzo per ridurre al minimo la possibilità di problemi con i client o errori di backup.

Connessioni terminate dal lato server

L'impostazione predefinita ElastiCache per la configurazione Redis mantiene le connessioni client stabilite a tempo indeterminato. Tuttavia, in alcuni casi la terminazione della connessione potrebbe essere auspicabile. Esempio:

- I bug nell'applicazione client possono causare connessioni da dimenticare e mantenere stabilite con uno stato di inattività. Questo è chiamato "perdita di connessione" e la conseguenza è un costante aumento del numero di connessioni stabilite osservate sul parametro `CurrentConnections`. Questo comportamento può comportare una saturazione del client o del lato. ElastiCache Quando una correzione immediata non è possibile dal lato client, alcuni amministratori impostano un valore di «timeout» nel proprio ElastiCache gruppo di parametri. Il timeout è il tempo in secondi consentito per la persistenza delle connessioni inattive. Se il client non invia alcuna

richiesta nel periodo, il motore Redis interromperà la connessione non appena la connessione raggiunge il valore di timeout. Piccoli valori di timeout possono causare disconnessioni non necessarie e i client dovranno gestirle correttamente e riconnettersi, causando ritardi.

- La memoria utilizzata per memorizzare le chiavi è condivisa con i buffer client. I client lenti con richieste o risposte di grandi dimensioni possono richiedere una quantità significativa di memoria per gestire i buffer. L'impostazione predefinita ElastiCache per le configurazioni Redis non limita la dimensione dei normali buffer di output del client. Se il limite `maxmemory` è raggiunto, il motore tenterà di espellere gli elementi per soddisfare l'utilizzo del buffer. In condizioni di memoria estremamente ridotta, ElastiCache For Redis potrebbe scegliere di disconnettere i client che consumano buffer di output client di grandi dimensioni per liberare memoria e mantenere lo stato del cluster.

È possibile limitare la dimensione dei buffer client con configurazioni personalizzate e i client che raggiungono il limite verranno disconnessi. Tuttavia, i client dovrebbero essere in grado di gestire disconnessioni impreviste. I parametri per gestire le dimensioni dei buffer per i client regolari sono i seguenti:

- `client-query-buffer-limit`: dimensione massima di una singola richiesta di input;
- `client-output-buffer-limit-normal-soft-limit`: Limite flessibile per le connessioni client. La connessione verrà interrotta se rimane al di sopra del limite libero per più del tempo in secondi definito su `client-output-buffer-limit-normal-soft-seconds` o se raggiunge il limite rigido;
- `client-output-buffer-limit-normal-soft-seconds`: Tempo consentito per le connessioni che superano il `client-output-buffer-limit-normal-soft-limit`;
- `client-output-buffer-limit-normal-hard-limit`: Una connessione che raggiunge questo limite verrà immediatamente interrotta.

Oltre ai buffer client normali, le seguenti opzioni controllano il buffer per i nodi di replica e i client Pub/Sub (Publish/Subscribe):

- `client-output-buffer-limit-replica-hard-limit`;
- `client-output-buffer-limit-replica-soft-seconds`;
- `client-output-buffer-limit-replica-hard-limit`;
- `client-output-buffer-limit-pubsub-soft-limit`;
- `client-output-buffer-limit-pubsub-soft-seconds`;
- `client-output-buffer-limit-pubsub-hard-limit`;

Risoluzione dei problemi lato client per istanze Amazon EC2

Il carico e la reattività sul lato client possono influire anche sulle richieste di ElastiCache. I limiti dell'istanza EC2 e del sistema operativo devono essere attentamente esaminati durante la risoluzione dei problemi di connettività intermittente o timeout. Alcuni punti chiave da osservare:

- CPU:
 - Utilizzo della CPU dell'istanza EC2: assicurarsi che la CPU non sia stata saturata o vicina al 100 per cento. L'analisi storica può essere effettuata tramite CloudWatch, tuttavia, tieni presente che la granularità dei punti dati è di 1 minuto (con il monitoraggio dettagliato abilitato) o di 5 minuti;
 - Se si utilizza [istanze EC2 burstable](#), assicurarsi che il loro saldo di credito della CPU non sia stato esaurito. Queste informazioni sono disponibili sulla metrica `CPUCreditBalance` CloudWatch
 - Brevi periodi di utilizzo elevato della CPU possono causare timeout senza che ciò comporti un utilizzo al 100%. CloudWatch Tali casi richiedono un monitoraggio in tempo reale con strumenti del sistema operativo come `top`, `ps` e `mpstat`.
- Rete
 - Verificare se la velocità effettiva di rete è inferiore ai valori accettabili in base alle capacità dell'istanza. Per ulteriori informazioni consultare [Tipi di istanza Amazon EC2](#)
 - Sulle istanze con ena Driver di rete avanzato, controllare le [Statistiche ENA](#) per i timeout o i limiti superati. Le seguenti statistiche sono utili per verificare la saturazione dei limiti di rete:
 - `bw_in_allowance_exceeded / bw_out_allowance_exceeded`: numero di pacchetti sagomati a causa di una eccessiva velocità effettiva in entrata o in uscita;
 - `conntrack_allowance_exceeded`: numero di pacchetti eliminati a causa dei [Limiti di monitoraggio delle connessioni](#) dei gruppi di sicurezza. Le nuove connessioni avranno esito negativo quando questo limite è saturo;
 - `linklocal_allowance_exceeded`: numero di pacchetti eliminati a causa di richieste eccessive per l'istanza di meta-dati, NTP tramite DNS VPC. Il limite è di 1024 pacchetti al secondo per tutti i servizi;
 - `pps_allowance_exceeded`: numero di pacchetti eliminati a causa di un rapporto eccessivo di pacchetti al secondo. Il limite PPS può essere raggiunto quando il traffico di rete è costituito da migliaia o milioni di richieste molto piccole al secondo. ElastiCache il traffico può essere ottimizzato per utilizzare meglio i pacchetti di rete tramite pipeline o comandi che eseguono più operazioni contemporaneamente, come invece di `MGET GET`

Dissezione del tempo necessario per completare una singola richiesta

- In rete: Tcpcmdump and Wireshark (tshark sulla riga di comando) sono strumenti utili per capire quanto tempo ha impiegato la richiesta per percorrere la rete, avviare il ElastiCache motore e ottenere un ritorno. L'esempio seguente evidenzia una singola richiesta creata con il seguente comando:

```
$ echo ping | nc example.xxxxxx.ng.0001.use1.cache.amazonaws.com 6379
+PONG
```

In parallelo al comando di cui sopra, tcpcmdump era in esecuzione ed è stato restituito:

```
$ sudo tcpcmdump -i any -nn port 6379 -tt
tcpcmdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
1609428918.917869 IP 172.31.11.142.40966
    > 172.31.11.247.6379: Flags [S], seq 177032944, win 26883, options [mss
    8961,sackOK,TS val 27819440 ecr 0,nop,wscale 7], length 0
1609428918.918071 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [S.], seq
    53962565, ack 177032945, win
    28960, options [mss 1460,sackOK,TS val 3788576332 ecr 27819440,nop,wscale 7],
    length 0
1609428918.918091 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 1, win
    211, options [nop,nop,TS val 27819440 ecr 3788576332], length 0
1609428918.918122
    IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [P.], seq 1:6, ack 1, win 211,
    options [nop,nop,TS val 27819440 ecr 3788576332], length 5: RESP "ping"
1609428918.918132 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [F.], seq 6, ack
    1, win 211, options [nop,nop,TS val 27819440 ecr 3788576332], length 0
1609428918.918240 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [.], ack 6, win
    227, options [nop,nop,TS val 3788576332 ecr 27819440], length 0
1609428918.918295
    IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [P.], seq 1:8, ack 7, win 227,
    options [nop,nop,TS val 3788576332 ecr 27819440], length 7: RESP "PONG"
1609428918.918300 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 8, win
    211, options [nop,nop,TS val 27819441 ecr 3788576332], length 0
1609428918.918302 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [F.], seq 8, ack
    7, win 227, options [nop,nop,TS val 3788576332 ecr 27819440], length 0
1609428918.918307
    IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.], ack 9, win 211, options
    [nop,nop,TS val 27819441 ecr 3788576332], length 0
^C
```

```
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

Dall'output di cui sopra possiamo confermare che l'handshake a tre vie TCP è stato completato in 222 microsecondi (918091 - 917869) e che il comando ping è stato inviato e restituito in 173 microsecondi (918295 - 918122).

Ci sono voluti 438 microsecondi (918307 - 917869) dalla richiesta alla chiusura della connessione. Tali risultati confermerebbero che i tempi di risposta della rete e del motore sono buoni e l'indagine può concentrarsi su altri componenti.

- Nel sistema operativo: `Strace` può aiutare a identificare le lacune temporali a livello di sistema operativo. L'analisi delle applicazioni effettive sarebbe molto più ampia e sono consigliati profiler di applicazioni specializzati o debugger. L'esempio seguente mostra solo se i componenti del sistema operativo di base funzionano come previsto, in caso contrario potrebbero essere necessarie ulteriori indagini. Utilizzando lo stesso comando PING di Redis con `strace` otteniamo:

```
$ echo ping | strace -f -tttt -r -e trace=execve,socket,open,recvfrom,sendto
nc example.xxxxxx.ng.0001.use1.cache.amazonaws.com (http://
example.xxxxxx.ng.0001.use1.cache.amazonaws.com/)
 6379
1609430221.697712 (+ 0.000000) execve("/usr/bin/nc", ["nc",
"example.xxxxxx.ng.0001.use"..., "6379"], 0x7ffffede7cc38 /* 22 vars */) = 0
1609430221.708955 (+ 0.011231) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|
SOCK_NONBLOCK, 0) = 3
1609430221.709084
(+ 0.000124) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK, 0) = 3
1609430221.709258 (+ 0.000173) open("/etc/nsswitch.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709637 (+ 0.000378) open("/etc/host.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709923
(+ 0.000286) open("/etc/resolv.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.711365 (+ 0.001443) open("/etc/hosts", O_RDONLY|O_CLOEXEC) = 3
1609430221.713293 (+ 0.001928) socket(AF_INET, SOCK_DGRAM|SOCK_CLOEXEC|SOCK_NONBLOCK,
IPPROTO_IP) = 3
1609430221.717419
(+ 0.004126) recvfrom(3, "\362|
\201\200\0\1\0\2\0\0\0\0\rnotls20201224\6tihew"..., 2048, 0, {sa_family=AF_INET,
sin_port=htons(53), sin_addr=inet_addr("172.31.0.2")}, [28->16]) = 155
1609430221.717890 (+ 0.000469) recvfrom(3,
"\204\207\201\200\0\1\0\1\0\0\0\0\rnotls20201224\6tihew"...,
```

```

65536, 0, {sa_family=AF_INET, sin_port=htons(53),
sin_addr=inet_addr("172.31.0.2")}, [28->16]) = 139
1609430221.745659 (+ 0.027772) socket(AF_INET, SOCK_STREAM, IPPROTO_TCP) = 3
1609430221.747548 (+ 0.001887) recvfrom(0, 0x7ffcf2f2ca50, 8192,
0, 0x7ffcf2f2c9d0, [128]) = -1 ENOTSOCK (Socket operation on non-socket)
1609430221.747858 (+ 0.000308) sendto(3, "ping\n", 5, 0, NULL, 0) = 5
1609430221.748048 (+ 0.000188) recvfrom(0, 0x7ffcf2f2ca50, 8192, 0, 0x7ffcf2f2c9d0,
[128]) = -1 ENOTSOCK
(Socket operation on non-socket)
1609430221.748330 (+ 0.000282) recvfrom(3, "+PONG\r\n", 8192, 0, 0x7ffcf2f2c9d0,
[128->0]) = 7
+PONG
1609430221.748543 (+ 0.000213) recvfrom(3, "", 8192, 0, 0x7ffcf2f2c9d0, [128->0]) = 0
1609430221.752110
(+ 0.003569) +++ exited with 0 +++

```

Nell'esempio precedente, il comando ha impiegato poco più di 54 millisecondi per essere completato (752110 - 697712 = 54398 microsecondi).

Una quantità significativa di tempo, circa 20 ms, è stata impiegata per istanziare nc ed effettuare la risoluzione dei nomi (da 697712 a 717890), dopodiché sono stati necessari 2 ms per creare il socket TCP (da 745659 a 747858) e 0,4 ms (da 747858 a 748330) per inviare e ricevere la risposta per la richiesta.

Sicurezza in Amazon ElastiCache

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il [modello di responsabilità condivisa](#) descrive questo come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori di terze parti testano e verificano regolarmente l'efficacia della sicurezza come parte dei [programmi di conformitàAWS](#). Per maggiori informazioni sui programmi di conformità applicabili ad Amazon ElastiCache, consulta [AWS Services in Scope by Compliance Program](#).
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa quando usi Amazon ElastiCache. I seguenti argomenti mostrano come configurare Amazon per ElastiCache soddisfare i tuoi obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue ElastiCache risorse Amazon.

Argomenti

- [Protezione dei dati in Amazon ElastiCache](#)
- [Riservatezza del traffico Internet](#)
- [Identity and Access Management per Amazon ElastiCache](#)
- [Convalida della conformità per Amazon ElastiCache](#)
- [Resilienza in Amazon ElastiCache](#)
- [Sicurezza dell'infrastruttura in AWS ElastiCache](#)
- [Aggiornamenti del servizio in ElastiCache](#)
- [Vulnerabilità ed esposizioni comuni \(CVE\): vulnerabilità di sicurezza risolte in per Redis ElastiCache](#)

Protezione dei dati in Amazon ElastiCache

Il [modello di responsabilità condivisa](#) AWS si applica alla protezione dei dati in AWS ElastiCache. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che esegue l'intero cloud AWS. L'utente è responsabile di mantenere il controllo sui contenuti ospitati su questa infrastruttura. Questo contenuto include la configurazione della protezione e le attività di gestione per i servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, consulta [Domande frequenti sulla privacy dei dati](#).

Per garantire la protezione dei dati, si consiglia di proteggere le credenziali dell'account AWS e di configurare singoli account con AWS Identity and Access Management (IAM). In questo modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere il proprio lavoro. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Utilizza TLS per comunicare con risorse AWS.
- Configura la creazione di log delle attività di API e utenti con AWS CloudTrail.
- Utilizza le soluzioni di crittografia AWS, insieme a tutti i controlli di sicurezza predefiniti all'interno dei servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, ad esempio Amazon Macie, che aiutano a individuare e proteggere i dati personali archiviati in Amazon S3.

Ti consigliamo di non inserire mai informazioni identificative sensibili, ad esempio i numeri di account dei clienti, in campi a formato libero, ad esempio un campo Name (Nome). Questo suggerimento si applica all'utilizzo di ElastiCache o altri servizi AWS con la console, l'API, la AWS CLI o gli SDK AWS. Gli eventuali dati immessi in ElastiCache o altri servizi possono essere prelevati per l'inserimento nei log di diagnostica. Quando fornisci un URL a un server esterno, non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta a tale server.

Argomenti

- [Sicurezza dei dati in Amazon ElastiCache](#)

Sicurezza dei dati in Amazon ElastiCache

Per mantenere la sicurezza dei dati, Amazon ElastiCache e Amazon EC2 forniscono meccanismi di protezione da accessi non autorizzati dei dati sul server.

Amazon ElastiCache per Redis fornisce funzionalità di crittografia per i dati su cache che eseguono Redis 3.2.6 (pianificato per la fine del ciclo di vita, consulta [Pianificazione della fine del ciclo di vita delle versioni di Redis](#)), 4.0.10 o versioni successive:

- La crittografia dei dati in transito viene eseguita quando i dati si spostano da una posizione a un'altra, ad esempio tra i nodi del cluster o tra la cache e l'applicazione.
- La crittografia dei dati inattivi esegue la crittografia su disco dei dati durante operazioni di sincronizzazione e backup.

Amazon ElastiCache per Redis supporta anche l'autenticazione degli utenti con IAM o Redis AUTH e l'autorizzazione delle operazioni degli utenti tramite il controllo degli accessi basato su ruoli (RBAC).

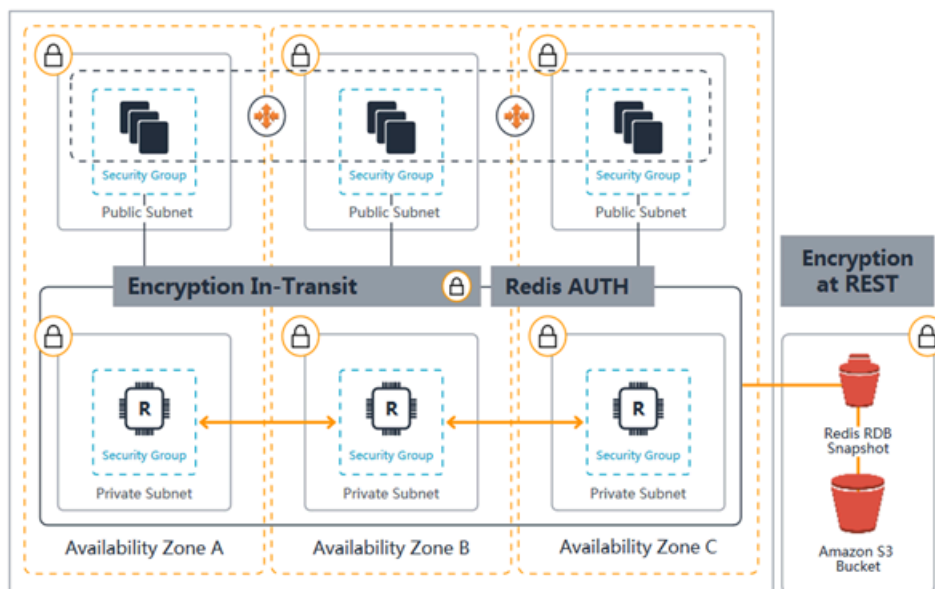


Diagramma di protezione ElastiCache

Argomenti

- [Crittografia in transito ElastiCache \(TLS\)](#)
- [Crittografia dei dati a riposo in ElastiCache](#)
- [Autenticazione e autorizzazione](#)

Crittografia in transito ElastiCache (TLS)

Per mantenere la sicurezza dei dati, Amazon ElastiCache e Amazon EC2 forniscono meccanismi di protezione da accessi non autorizzati dei dati sul server. Grazie alla caratteristica di crittografia dei

dati in transito fornita da ElastiCache, disponi di uno strumento che puoi utilizzare per proteggere i dati quando si spostano da una posizione a un'altra.

Tutte le cache serverless dispongono della crittografia dei dati in transito abilitata. Per i cluster progettati autonomamente puoi abilitare la crittografia dei dati in transito su un gruppo di replica impostando il parametro `TransitEncryptionEnabled` su `true` (CLI: `--transit-encryption-enabled`) durante la creazione del gruppo di replica. Ciò è possibile se crei il gruppo di replica utilizzando la AWS Management Console, AWS CLI o l'API ElastiCache.

Argomenti

- [Panoramica della crittografia dei dati in transito](#)
- [Condizioni di crittografia dei dati in transito](#)
- [Best practice per la crittografia in transito](#)
- [Consulta anche](#)
- [Abilitazione della crittografia dei dati in transito](#)
- [Connessione ad Amazon ElastiCache for Redis con crittografia in transito tramite redis-cli](#)
- [Abilitazione della crittografia dei dati in transito su un cluster Redis progettato autonomamente mediante Python](#)
- [Best practice per abilitare la crittografia dei dati in transito](#)

Panoramica della crittografia dei dati in transito

La crittografia dei dati in transito di Amazon ElastiCache è una funzionalità che ti consente di aumentare la sicurezza dei tuoi dati nei punti più vulnerabili, quando sono in transito da una posizione all'altra. Poiché la crittografia e la decrittografia dei dati richiede l'elaborazione a livello di endpoint, l'abilitazione della crittografia dei dati in transito può in parte influire sulle prestazioni. È opportuno creare un riferimento per i dati con o senza crittografia dei dati in transito per determinare l'impatto sulle prestazioni per i propri casi d'uso.

La crittografia dei dati in transito ElastiCache implementa le seguenti caratteristiche:

- Connessioni client crittografate: le connessioni client ai nodi di cache sono crittografate con TLS.
- Connessioni server crittografate: i dati che si spostano tra i nodi di un cluster sono crittografati.
- Autenticazione del server: i client possono autenticare che si stanno connettendo al server giusto.
- Autenticazione client, utilizzando la caratteristica Redis AUTH, il server può autenticare i client.

Condizioni di crittografia dei dati in transito

Tieni presente i seguenti vincoli sulla crittografia dei dati in transito di Amazon ElastiCache durante la pianificazione dell'implementazione di un cluster progettato autonomamente:

- La crittografia dei dati in transito è supportata su gruppi di replica che eseguono Redis versione 3.2.6, 4.0.10 e successive.
- La modifica dell'impostazione di crittografia dei dati in transito, per un cluster esistente, è supportata nei gruppi di replica che eseguono Redis versione 7 e successive.
- La crittografia dei dati in transito è supportata solo per gruppi di replica in esecuzione in un Amazon VPC.
- La crittografia dei dati in transito è supportata solo per i gruppi di replica in esecuzione sui seguenti tipi di nodi.
 - R6g, R5, R4, R3
 - M6g, M5, M4, M3
 - T4g, T3, T2

Per ulteriori informazioni, consulta [Tipi di nodi supportati](#).

- La crittografia dei dati in transito è abilitata impostando in maniera esplicita il parametro `TransitEncryptionEnabled` su `true`.
- Assicurati che il client di cache supporti la connettività TLS e che sia abilitata nella configurazione del client.

Best practice per la crittografia in transito

- A causa dell'elaborazione richiesta per crittografare e decrittografare i dati a livello degli endpoint, l'implementazione della crittografia dei dati in transito può ridurre le prestazioni. Raffronta la crittografia dei dati in transito con l'assenza di crittografia sui dati per determinare il suo impatto sulle prestazioni per l'implementazione.
- Siccome la creazione di nuove connessioni può richiedere molte risorse, puoi ridurre l'impatto della crittografia in transito sulle prestazioni mantenendo le connessioni SSL.

Consulta anche

- [Crittografia dei dati a riposo in ElastiCache](#)
- [Autenticazione con il comando Redis AUTH](#)

- [Autenticazione degli utenti con il controllo degli accessi basato sui ruoli \(RBAC\)](#)
- [Amazon VPC e sicurezza di ElastiCache](#)
- [Identity and Access Management per Amazon ElastiCache](#)

Abilitazione della crittografia dei dati in transito

Tutte le cache serverless dispongono della crittografia dei dati in transito abilitata. In un cluster progettato autonomamente puoi abilitare la crittografia dei dati in transito mediante la AWS Management Console, l'interfaccia della linea di comando AWS o l'API ElastiCache.

Abilitazione della crittografia dei dati in transito mediante la AWS Management Console

Abilitazione della crittografia dei dati in transito per un nuovo cluster progettato autonomamente mediante la AWS Management Console

Quando si progetta il proprio cluster, le configurazioni "Sviluppo/Test" e "Produzione" con il metodo "Crea semplice" hanno la crittografia dei dati in transito abilitata. Quando scegli la configurazione, procedi come segue:

- Scegli la versione motore 3.2.6, 4.0.10 o successiva.
- Fai clic sulla casella di controllo Abilita accanto all'opzione Crittografia dei dati in transito.

Per il processo dettagliato, consulta la seguente documentazione:

- [Creazione di un cluster Redis \(modalità cluster disabilitata\) \(Console\)](#)
- [Creazione di un cluster Redis \(modalità cluster abilitata\) \(console\).](#)

Abilitazione della crittografia dei dati in transito per un cluster progettato autonomamente esistente mediante la AWS Management Console

L'abilitazione della crittografia dei dati in transito è un processo in due fasi; imposta innanzitutto la modalità di crittografia dei dati in transito su `preferred`. Questa modalità consente ai client Redis di connettersi utilizzando connessioni crittografate e non crittografate. Dopo aver eseguito la migrazione di tutti i client Redis per utilizzare connessioni crittografate, puoi modificare la configurazione del cluster per impostare la modalità di crittografia dei dati in transito su `required`. L'impostazione della modalità di crittografia dei dati in transito su `required` rimuoverà tutte le connessioni non crittografate e consentirà solo le connessioni crittografate.

Fase 1: imposta Transit encryption mode (Modalità crittografia in transito) su Preferred (Preferito)

1. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Scegli Cache Redis dalle risorse ElastiCache elencate nel pannello di navigazione, presente sulla sinistra.
3. Scegli la cache Redis da aggiornare.
4. Scegli il menu a discesa Actions (Azioni), quindi Modify (Modifica).
5. Scegli Enable (Abilita) in Encryption in transit (Crittografia dei dati in transito) nella sezione Security (Sicurezza).
6. Scegli Preferred (Preferito) come Transit encryption mode (Modalità di crittografia dei dati in transito).
7. Scegli Preview changes (Anteprima modifiche) e salva le modifiche.

Dopo aver eseguito la migrazione di tutti i client Redis per utilizzare connessioni crittografate:

Fase 2: imposta Transit encryption mode (Modalità crittografia dati in transito) su Required (Richiesto)

1. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Scegli Cache Redis dalle risorse ElastiCache elencate nel pannello di navigazione, presente sulla sinistra.
3. Scegli la cache Redis da aggiornare.
4. Scegli il menu a discesa Actions (Azioni), quindi Modify (Modifica).
5. Scegli Required (Richiesto) come Transit encryption mode (Modalità di crittografia dei dati in transito), nella sezione Security (Sicurezza).
6. Scegli Preview changes (Anteprima modifiche) e salva le modifiche.

Abilitazione della crittografia dei dati in transito mediante la AWS CLI

Per abilitare la crittografia dei dati in transito creando un gruppo di replica Redis mediante AWS CLI, utilizza il parametro `transit-encryption-enabled`.

Abilitazione della crittografia dei dati in transito su un nuovo cluster progettato autonomamente per Redis (modalità cluster disabilitata) (CLI)

Utilizza l'operazione `create-replication-group` di AWS CLI e i seguenti parametri per creare un gruppo di replica Redis con le repliche in cui la crittografia dei dati in transito è abilitata:

Parametri chiave:

- **--engine** — Deve essere `redis`.
- **--engine-version**-Deve essere 3.2.6, 4.0.10 o versione successiva.
- **--transit-encryption-enabled**: obbligatorio. Se abiliti la crittografia dei dati in transito, devi anche fornire un valore per il parametro `--cache-subnet-group`.
- **--num-cache-clusters**-deve essere almeno 1. Il valore massimo per questo parametro è sei.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Creazione da zero di un gruppo di replica Redis \(modalità cluster disabilitata\) \(AWS CLI\)](#)
- [create-replication-group](#)

Abilitazione della crittografia dei dati in transito su un nuovo cluster progettato autonomamente per Redis (modalità cluster abilitata) (CLI)

Utilizza l'operazione `create-replication-group` di AWS CLI e i seguenti parametri per creare un gruppo di replica Redis (modalità cluster attivata) con le repliche in cui la crittografia dei dati in transito è abilitata:

Parametri chiave:

- **--engine** — Deve essere `redis`.
- **--engine-version**-Deve essere 3.2.6, 4.0.10 o versione successiva.
- **--transit-encryption-enabled**: obbligatorio. Se abiliti la crittografia dei dati in transito, devi anche fornire un valore per il parametro `--cache-subnet-group`.
- Utilizza uno dei seguenti set di parametri per specificare la configurazione dei gruppi di nodi del gruppo di replica:
 - **--num-node-groups**-specifica il numero di partizioni (gruppi di nodi) in questo gruppo di replica. Il valore massimo di questo parametro è 500.

- **--replicas-per-node-group**—specifica il numero di nodi di replica in ogni gruppo di nodi. Il valore specificato qui viene applicato a tutte le partizioni in questo gruppo di replica. Il valore massimo di questo parametro è 5.
- **--node-group-configuration**- Specifica la configurazione di ogni partizione in modo indipendente.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Creazione da zero di un gruppo di replica Redis \(modalità cluster abilitata\) \(AWS CLI\)](#)
- [create-replication-group](#)

Abilitazione della crittografia dei dati in transito per un cluster esistente mediante la AWS CLI

L'abilitazione della crittografia dei dati in transito è un processo in due fasi; imposta innanzitutto la modalità di crittografia dei dati in transito su `preferred`. Questa modalità consente ai client Redis di connettersi utilizzando connessioni crittografate e non crittografate. Dopo aver eseguito la migrazione di tutti i client Redis per utilizzare connessioni crittografate, puoi modificare la configurazione del cluster per impostare la modalità di crittografia dei dati in transito su `required`. L'impostazione della modalità di crittografia dei dati in transito su `required` rimuoverà tutte le connessioni non crittografate e consentirà solo le connessioni crittografate.

Utilizza l'operazione AWS CLI `modify-replication-group` e i seguenti parametri per aggiornare un gruppo di replica Redis (modalità cluster attivata) con le repliche in cui la crittografia dei dati in transito è disabilitata.

Per abilitare la crittografia dei dati in transito

1. Imposta `transit-encryption-mode` su `preferred`, utilizzando i seguenti parametri
 - **--transit-encryption-enabled**: obbligatorio.
 - **--transit-encryption-mode** – Deve essere impostato su `preferred`.
2. Imposta `transit-encryption-mode` su `required`, utilizzando i seguenti parametri:
 - **--transit-encryption-enabled**: obbligatorio.
 - **--transit-encryption-mode** – Deve essere impostato su `required`.

Connessione ad Amazon ElastiCache for Redis con crittografia in transito tramite redis-cli

Per accedere ai dati provenienti da ElastiCache cache Redis abilitate con crittografia in transito, 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. Se il client non supporta TLS, è possibile utilizzare il comando `stunnel` sull'host client per creare un tunnel SSL ai nodi Redis.

Connessione crittografata con Linux

Per utilizzare redis-cli per connettersi a un cluster Redis abilitato con crittografia in transito su Amazon Linux 2023, Amazon Linux 2 o Amazon Linux, segui questi passaggi.

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
```

Connessione crittografata con stunnel

Per utilizzare redis-cli per connetterti a un cluster Redis abilitato con crittografia in transito tramite stunnel, segui questi passaggi.

1. Utilizza SSH per connettersi al client e installare stunnel.

```
sudo yum install stunnel
```

2. Esegui il comando seguente per creare e modificare il file '/etc/stunnel/redis-cli.conf' contemporaneamente per aggiungere un endpoint del cluster ElastiCache for Redis a uno o più parametri di connessione, utilizzando l'output fornito di seguito come modello.

```
vi /etc/stunnel/redis-cli.conf

fips = no
setuid = root
setgid = root
pid = /var/run/stunnel.pid
debug = 7
delay = yes
options = NO_SSLv2
options = NO_SSLv3
[redis-cli]
  client = yes
  accept = 127.0.0.1:6379
  connect = primary.ssltest.wif01h.use1.cache.amazonaws.com:6379
[redis-cli-replica]
  client = yes
  accept = 127.0.0.1:6380
  connect = ssltest-02.ssltest.wif01h.use1.cache.amazonaws.com:6379
```

In questo esempio, il file di configurazione dispone di due connessioni, `redis-cli` e `redis-cli-replica`. I parametri sono impostati come riportato di seguito:

- `client` è impostato su `yes` per specificare che questa istanza `stunnel` è un client.
- `accept` è impostato sull'IP client. In questo esempio, il master è impostato sul valore di default Redis 127.0.0.1 sulla porta 6379. La replica deve chiamare una porta diversa e impostarla su 6380. Puoi usare porte effimere 1024–65535. Per ulteriori informazioni, consulta [Porte Effimere](#) nella guida per l'utente Amazon VPC.
- `connect` è impostato sull'endpoint del server Redis. Per ulteriori informazioni, consulta [Individuazione degli endpoint di connessione](#).

3. Avvia stunnel.

```
sudo stunnel /etc/stunnel/redis-cli.conf
```

Utilizza il comando `netstat` per confermare che i tunnel sono stati avviati.

```
sudo netstat -tulnp | grep -i stunnel

tcp        0      0 127.0.0.1:6379        0.0.0.0:*             LISTEN
           3189/stunnel
tcp        0      0 127.0.0.1:6380        0.0.0.0:*             LISTEN
           3189/stunnel
```

4. Connettiti al nodo Redis crittografato utilizzando l'endpoint locale del tunnel.

- Se non è stata utilizzata alcuna password AUTH durante la ElastiCache creazione del cluster Redis, questo esempio utilizza `redis-cli` per connettersi al server ElastiCache for Redis utilizzando il percorso completo per `redis-cli`, su Amazon Linux:

```
/home/ec2-user/redis-stable/src/redis-cli -h localhost -p 6379
```

Se durante la creazione del cluster è stata utilizzata una password AUTH, questo esempio utilizza `redis-cli` per connettersi al server Redis utilizzando il percorso completo per `redis-cli`, su Amazon Linux:

```
/home/ec2-user/redis-stable/src/redis-cli -h localhost -p 6379 -a my-secret-password
```

O

- Modifica la directory su `redis-stable` ed effettua le seguenti operazioni:

Se non è stata utilizzata alcuna password AUTH durante la ElastiCache creazione del cluster Redis, questo esempio utilizza `redis-cli` per connettersi al server ElastiCache for Redis utilizzando il percorso completo per `redis-cli`, su Amazon Linux:

```
src/redis-cli -h localhost -p 6379
```

Se durante la creazione del cluster è stata utilizzata una password AUTH, questo esempio utilizza `redis-cli` per connettersi al server Redis utilizzando il percorso completo per `redis-cli`, su Amazon Linux:

```
src/redis-cli -h localhost -p 6379 -a my-secret-password
```

Questo esempio utilizza Telnet per connettersi al server Redis.

```
telnet localhost 6379

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
auth MySecretPassword
+OK
get foo
$3
bar
```

5. Per interrompere e chiudere i tunnel SSL, esegui il `kill` del processo `stunnel`.

```
sudo kill stunnel
```

Abilitazione della crittografia dei dati in transito su un cluster Redis progettato autonomamente mediante Python

Nella guida seguente viene illustrato come abilitare la crittografia dei dati in transito su un cluster Redis 7.0 originariamente creato con la crittografia dei dati in transito disabilitata. I client TCP e TLS continueranno a comunicare con il cluster durante questo processo senza tempi di inattività.

Boto3 otterrà le credenziali necessarie (`aws_access_key_id`, `aws_secret_access_key` e `aws_session_token`) dalle variabili di ambiente. Tali credenziali verranno incollate in anticipo nello stesso terminale bash in cui verrà eseguito `python3` per elaborare il codice Python mostrato in questa guida. Il codice nell'esempio sottostante è stato elaborato da un'istanza EC2 avviata nello stesso VPC che verrà utilizzato per creare il cluster Redis ElastiCache al suo interno.

Note

- Gli esempi seguenti utilizzano l'SDK boto3 per le operazioni di gestione ElastiCache (creazione di cluster o utenti) e redis-py/redis-py-cluster per la gestione dei dati.
- È necessario utilizzare almeno la versione boto3 (=~) 1.26.39 per utilizzare la migrazione TLS online con l'API di modifica del cluster.
- ElastiCache supporta la migrazione TLS online solo per i cluster Redis con versione 7.0 o successiva. Pertanto, se si dispone di un cluster che esegue una versione Redis precedente alla 7.0, sarà necessario aggiornare la versione Redis del cluster. Per ulteriori informazioni sulle differenze di versione, consultare [Differenze di compatibilità e comportamento delle versioni principali](#).

Argomenti

- [Definizione delle costanti stringa che avvieranno il cluster Redis ElastiCache](#)
- [Definizione delle classi per la configurazione del cluster](#)
- [Definizione di una classe che rappresenterà il cluster stesso](#)
- [\(Facoltativo\) Creazione di una classe wrapper per dimostrare la connessione client al cluster Redis](#)
- [Creazione della funzione principale che illustra il processo di modifica della configurazione della crittografia dei dati in transito](#)

Definizione delle costanti stringa che avvieranno il cluster Redis ElastiCache

Innanzitutto, definiamo alcune semplici costanti stringa Python che conterranno i nomi delle entità AWS necessarie per creare il cluster ElastiCache come `security-group`, `Cache Subnet group` e un `default parameter group`. Tutte queste entità AWS devono essere create in anticipo nell'account AWS nella regione che si intende utilizzare.

```
#Constants definitions
SECURITY_GROUP = "sg-0492aa0a29c558427"
CLUSTER_DESCRIPTION = "This cluster has been launched as part of the online TLS
migration user guide"
EC_SUBNET_GROUP = "client-testing"
DEFAULT_PARAMETER_GROUP_REDIS_7_CLUSTER_MODE_ENABLED = "default.redis7.cluster.on"
```

Definizione delle classi per la configurazione del cluster

Ora, definiamo alcune semplici classi Python che rappresentano una configurazione di un cluster, che conterranno i metadati relativi al cluster come la versione Redis, il tipo di istanza e la specifica se la crittografia dei dati in transito (TLS) è abilitata o disabilitata.

```
#Class definitions

class Config:
    def __init__(
        self,
        instance_type: str = "cache.t4g.small",
        version: str = "7.0",
        multi_az: bool = True,
        TLS: bool = True,
        name: str = None,
    ):
        self.instance_type = instance_type
        self.version = version
        self.multi_az = multi_az
        self.TLS = TLS
        self.name = name or f"tls-test"

    def create_base_launch_request(self):
        return {
            "ReplicationGroupId": self.name,
            "TransitEncryptionEnabled": self.TLS,
            "MultiAZEnabled": self.multi_az,
            "CacheNodeType": self.instance_type,
            "Engine": "redis",
            "EngineVersion": self.version,
            "CacheSubnetGroupName": EC_SUBNET_GROUP ,
            "CacheParameterGroupName":
DEFAULT_PARAMETER_GROUP_REDIS_7_CLUSTER_MODE_ENABLED ,
            "ReplicationGroupDescription": CLUSTER_DESCRIPTION,
            "SecurityGroupIds": [SECURITY_GROUP],
        }

class ConfigCME(Config):
    def __init__(
        self,
        instance_type: str = "cache.t4g.small",
        version: str = "7.0",
```

```

    multi_az: bool = True,
    TLS: bool = True,
    name: str = None,
    num_shards: int = 2,
    num_replicas_per_shard: int = 1,
):
    super().__init__(instance_type, version, multi_az, TLS, name)
    self.num_shards = num_shards
    self.num_replicas_per_shard = num_replicas_per_shard

def create_launch_request(self) -> dict:
    launch_request = self.create_base_launch_request()
    launch_request["NumNodeGroups"] = self.num_shards
    launch_request["ReplicasPerNodeGroup"] = self.num_replicas_per_shard
    return launch_request

```

Definizione di una classe che rappresenterà il cluster stesso

Ora, definiamo alcune semplici classi Python che rappresenteranno il cluster Redis ElastiCache stesso. Questa classe avrà un campo client che conterrà un client boto3 per operazioni di gestione ElastiCache come la creazione del cluster e l'esecuzione di query sull'API ElastiCache.

```

import botocore.config
import boto3

# Create boto3 client
def init_client(region: str = "us-east-1"):
    config = botocore.config.Config(retries={"max_attempts": 10, "mode": "standard"})
    init_request = dict()
    init_request["config"] = config
    init_request["service_name"] = "elasticache"
    init_request["region_name"] = region
    return boto3.client(**init_request)

class ElastiCacheClusterBase:
    def __init__(self, name: str):
        self.name = name
        self.elasticache_client = init_client()

    def get_first_replication_group(self):
        return self.elasticache_client.describe_replication_groups(
            ReplicationGroupId=self.name

```

```
    )["ReplicationGroups"][0]

def get_status(self) -> str:
    return self.get_first_replication_group()["Status"]

def get_transit_encryption_enabled(self) -> bool:
    return self.get_first_replication_group()["TransitEncryptionEnabled"]

def is_available(self) -> bool:
    return self.get_status() == "available"

def is_modifying(self) -> bool:
    return self.get_status() == "modifying"

def wait_for_available(self):
    while True:
        if self.is_available():
            break
        else:
            time.sleep(5)

def wait_for_modifying(self):
    while True:
        if self.is_modifying():
            break
        else:
            time.sleep(5)

def delete_cluster(self) -> bool:
    self.elasticache_client.delete_replication_group(
        ReplicationGroupId=self.name, RetainPrimaryCluster=False
    )

def modify_transit_encryption_mode(self, new_transit_encryption_mode: str):
    # generate api call to migrate the cluster to TLS preferred or to TLS required
    self.elasticache_client.modify_replication_group(
        ReplicationGroupId=self.name,
        TransitEncryptionMode=new_transit_encryption_mode,
        TransitEncryptionEnabled=True,
        ApplyImmediately=True,
    )
    self.wait_for_modifying()

class ElastiCacheClusterCME(ElastiCacheClusterBase):
```

```
def __init__(self, name: str):
    super().__init__(name)

    @classmethod
    def launch(cls, config: ConfigCME = None) -> ElastiCacheClusterCME:
        config = config or ConfigCME()
        print(config)
        new_cluster = ElastiCacheClusterCME(config.name)
        launch_request = config.create_launch_request()
        new_cluster.elasticache_client.create_replication_group(**launch_request)
        new_cluster.wait_for_available()
        return new_cluster

    def get_configuration_endpoint(self) -> str:
        return self.get_first_replication_group()["ConfigurationEndpoint"]["Address"]

#Since the code can throw exceptions, we define this class to make the code more
#readable and
#so we won't forget to delete the cluster
class ElastiCacheCMEManager:
    def __init__(self, config: ConfigCME = None):
        self.config = config or ConfigCME()

    def __enter__(self) -> ElastiCacheClusterCME:
        self.cluster = ElastiCacheClusterCME.launch(self.config)
        return self.cluster

    def __exit__(self, exc_type, exc_val, exc_tb):
        self.cluster.delete_cluster()
```

(Facoltativo) Creazione di una classe wrapper per dimostrare la connessione client al cluster Redis

Ora, creiamo una classe wrapper per il client `redis-py-cluster`. Questa classe wrapper supporta la precompilazione del cluster con alcune chiavi e l'esecuzione di comandi `get` ripetuti in modo casuale.

Note

Questo è un passaggio facoltativo ma semplifica il codice della funzione principale che verrà fornito in un passaggio successivo.


```
import redis
import random
from time import perf_counter_ns, time

class DowntimeTestClient:
    def __init__(self, client):
        self.client = client

        # num of keys prefilled
        self.prefilled = 0
        # percent of get above prefilled
        self.percent_get_above_prefilled = 10 # nil result expected when get hit above
prefilled
        # total downtime in nano seconds
        self.downtime_ns = 0
        # num of success and fail operations
        self.success_ops = 0
        self.fail_ops = 0
        self.connection_errors = 0
        self.timeout_errors = 0

    def replace_client(self, client):
        self.client = client

    def prefill_data(self, timelimit_sec=60):
        end_time = time() + timelimit_sec
        while time() < end_time:
            self.client.set(self.prefilled, self.prefilled)
            self.prefilled += 1

    # unsuccessful operations throw exceptions
    def _exec(self, func):
        try:
            start_ns = perf_counter_ns()
            func()
            self.success_ops += 1
            elapsed_ms = (perf_counter_ns() - start_ns) // 10 ** 6
            # upon succesful execution of func
            # reset random_key to None so that the next command
            # will use a new random key
            self.random_key = None
```

```
except Exception as e:
    elapsed_ns = perf_counter_ns() - start_ns
    self.downtime_ns += elapsed_ns
    # in case of failure- increment the relevant counters so that we will keep
track
    # of how many connection issues we had while trying to communicate with
    # the cluster.
    self.fail_ops += 1
    if e.__class__ is redis.exceptions.ConnectionError:
        self.connection_errors += 1
    if e.__class__ is redis.exceptions.TimeoutError:
        self.timeout_errors += 1

def _repeat_exec(self, func, seconds):
    end_time = time() + seconds
    while time() < end_time:
        self._exec(func)

def _new_random_key_if_needed(self, percent_above_prefilled):
    if self.random_key is None:
        max = int((self.prefilled * (100 + percent_above_prefilled)) / 100)
        return random.randint(0, max)
    return self.random_key

def _random_get(self):
    key = self._new_random_key_if_needed(self.percent_get_above_prefilled)
    result = self.client.get(key)
    # we know the key was set for sure only in the case key < self.prefilled
    if key < self.prefilled:
        assert result.decode("UTF-8") == str(key)

def repeat_get(self, seconds=60):
    self._repeat_exec(self._random_get, seconds)

def get_downtime_ms(self) -> int:
    return self.downtime_ns // 10 ** 6

def do_get_until(self, cond_check):
    while not cond_check():
        self.repeat_get()
    # do one more get cycle once condition is met
```

```
self.repeat_get()
```

Creazione della funzione principale che illustra il processo di modifica della configurazione della crittografia dei dati in transito

Ora, definiamo la funzione principale, che eseguirà le operazioni seguenti:

1. Creare il cluster utilizzando il client ElastiCache boto3.
2. Inizializzare il client `redis-py-cluster` che si conatterà al cluster con una connessione TCP chiara senza TLS.
3. Il client `redis-py-cluster` precompila il cluster con alcuni dati.
4. Il client boto3 attiverà la migrazione TLS da no-TLS a TLS preferred.
5. Durante la migrazione del cluster a TLS Preferred, il client TCP `redis-py-cluster` invierà operazioni get ripetute al cluster finché la migrazione non è terminata.
6. Al termine della migrazione a TLS Preferred, affermeremo che il cluster supporta la crittografia dei dati in transito. Successivamente, creeremo un client `redis-py-cluster` che si conatterà al cluster con TLS.
7. Invieremo alcuni comandi get utilizzando il nuovo client TLS e il vecchio client TCP.
8. Il client boto3 attiverà la migrazione TLS da TLS Preferred a TLS required.
9. Durante la migrazione del cluster a TLS required, il client TCP `redis-py-cluster` invierà operazioni get ripetute al cluster finché la migrazione non è terminata.

```
import redis

def init_cluster_client(
    cluster: ElastiCacheClusterCME, prefill_data: bool, TLS: bool = True) ->
DowntimeTestClient:
    # we must use for the host name the cluster configuration endpoint.
    redis_client = redis.RedisCluster(
        host=cluster.get_configuration_endpoint(), ssl=TLS, socket_timeout=0.25,
        socket_connect_timeout=0.1
    )
    test_client = DowntimeTestClient(redis_client)
    if prefill_data:
        test_client.prefill_data()
    return test_client

if __name__ == '__main__':
```

```
config = ConfigCME(TLS=False, instance_type="cache.m5.large")

with ElastiCacheCMEManager(config) as cluster:
    # create a client that will connect to the cluster with clear tcp connection
    test_client_tcp = init_cluster_client(cluster, prefill_data=True, TLS=False)

    # migrate the cluster to TLS Preferred
    cluster.modify_transit_encryption_mode(new_transit_encryption_mode="preferred")

    # do repeated get commands until the cluster finishes the migration to TLS
    Preferred
    test_client_tcp.do_get_until(cluster.is_available)

    # verify that in transit encryption is enabled so that clients will be able to
    connect to the cluster with TLS
    assert cluster.get_transit_encryption_enabled() == True

    # create a client that will connect to the cluster with TLS connection.
    # we must first make sure that the cluster indeed supports TLS
    test_client_tls = init_cluster_client(cluster, prefill_data=True, TLS=True)

    # by doing get commands with the tcp client for 60 more seconds
    # we can verify that the existing tcp connection to the cluster still works
    test_client_tcp.repeat_get(seconds=60)

    # do get commands with the new TLS client for 60 more seconds
    test_client_tcp.repeat_get(seconds=60)

    # migrate the cluster to TLS required
    cluster.modify_transit_encryption_mode(new_transit_encryption_mode="required")

    # from this point the tcp clients will be disconnected and we must not use them
    anymore.
    # do get commands with the TLS client until the cluster finishes migration to
    TLS required mode.
    test_client_tls.do_get_until(cluster.is_available)
```

Best practice per abilitare la crittografia dei dati in transito

Prima di abilitare la crittografia dei dati in transito: accertarsi di disporre della gestione dei record DNS corretta

Note

Stiamo modificando ed eliminando i vecchi endpoint durante questo processo. L'uso errato degli endpoint può comportare l'utilizzo di endpoint vecchi ed eliminati che impediranno al client Redis di connettersi al cluster.

Durante la migrazione del cluster da no-TLS a TLS-preferred, i vecchi record DNS per nodo vengono conservati e i nuovi record DNS per nodo vengono generati in un formato diverso. I cluster abilitati per TLS utilizzano un formato di record DNS diverso rispetto ai cluster non abilitati per TLS. ElastiCache conserverà entrambi i record DNS quando un cluster è configurato in modalità di crittografia: Preferred in modo che le applicazioni e gli altri client Redis possano passare da uno all'altro. Le seguenti modifiche nei record DNS vengono apportate durante il processo di migrazione TLS:

Descrizione delle modifiche nei record DNS che vengono eseguite quando si abilita la crittografia dei dati in transito

Per i cluster CME

Quando un cluster è impostato sulla 'modalità di crittografia dei dati in transito: preferred':

- Gli endpoint del cluster originali per cluster non abilitati per TLS rimarranno attivi. Non ci saranno tempi di inattività quando il cluster viene riconfigurato dalla modalità di crittografia TLS 'none' a 'preferred'.
- I nuovi endpoint TLS Redis verranno generati quando il cluster è impostato sulla modalità TLS-preferred. Questi nuovi endpoint verranno risolti negli stessi IP di quelli vecchi (non TLS).
- Il nuovo endpoint di configurazione TLS Redis verrà esposto nella console ElastiCache e nella risposta all'API `describe-replication-group`.

Quando un cluster è impostato sulla 'modalità di crittografia dei dati in transito: required':

- I vecchi endpoint non abilitati per TLS verranno eliminati. Non ci saranno tempi di inattività degli endpoint del cluster TLS.

- Puoi recuperare un nuovo `cluster-configuration-endpoint` dalla console ElastiCache o dall'API `describe-replication-group`.

Per i cluster CMD con failover automatico abilitato o failover automatico disabilitato

Quando il gruppo di replica è impostato sulla 'modalità di crittografia dei dati in transito: preferred':

- L'endpoint primario e l'endpoint di lettura originali per i cluster non abilitati per TLS rimarranno attivi.
- I nuovi endpoint primari e di lettura TLS verranno generati quando il cluster è impostato sulla modalità TLS Preferred. Questo nuovo endpoint verrà risolto nello stesso IP di quelli vecchi (non-TLS).
- I nuovi endpoint primari e di lettura verranno esposti nella console ElastiCache e nella risposta all'API `describe-replication-group`.

Quando il gruppo di replica è impostato sulla 'modalità di crittografia dei dati in transito: required':

- L'endpoint primario e l'endpoint di lettura originali per i cluster non abilitati per TLS rimarranno attivi.
- I vecchi endpoint primari e di lettura non TLS verranno eliminati. Non ci saranno tempi di inattività degli endpoint del cluster TLS.
- Puoi recuperare nuovi endpoint primari e di lettura dalla console ElastiCache o dall'API `describe-replication-group`.

L'utilizzo suggerito dei record DNS

Per i cluster CME

- Utilizza l'endpoint di configurazione del cluster anziché i record DNS per nodo nel codice dell'applicazione. L'utilizzo diretto dei nomi DNS per nodo non è consigliato perché potrebbero cambiare quando si aggiungono o si rimuovono partizioni.
- Non codificare l'endpoint di configurazione del cluster nell'applicazione, poiché questo cambierà durante questo processo.
- Avere l'endpoint di configurazione del cluster codificato nell'applicazione è una bad practice poiché può essere modificato durante questo processo. Al termine della crittografia dei dati in transito, esegui query sull'endpoint di configurazione del cluster con l'API `describe-replication-group` (come illustrato sopra in grassetto) e utilizza il DNS che ricevi in risposta da questo punto in avanti.

Per cluster CMD con failover automatico abilitato

- Utilizza l'endpoint primario e l'endpoint di lettura anziché i nomi DNS per nodo nel codice dell'applicazione poiché i vecchi nomi DNS per nodo vengono eliminati e nuovi vengono generati durante la migrazione del cluster da no-TLS a TLS-preferred. L'utilizzo diretto dei nomi DNS per nodo non è consigliato perché è possibile che vengano aggiunte repliche al cluster in futuro. Inoltre, quando il failover automatico è abilitato, i ruoli del cluster primario e delle repliche vengono modificati automaticamente dal servizio ElastiCache. L'utilizzo dell'endpoint primario e dell'endpoint di lettura è consigliato per tenere traccia di tali modifiche. Infine, l'utilizzo dell'endpoint di lettura consente di distribuire le letture dalle repliche equamente tra le repliche nel cluster.
- Avere l'endpoint primario e l'endpoint di lettura codificati nell'applicazione è una bad practice poiché possono essere modificati durante il processo di migrazione TLS. Dopo che la modifica a TLS-preferred è stata completata, esegui query sull'endpoint primario e sull'endpoint di lettura con l'API describe-replication-group e utilizza il DNS che ricevi in risposta da questo punto in avanti. In questo modo sarai in grado di tenere traccia delle modifiche negli endpoint in modo dinamico.

Per cluster CMD con failover automatico disabilitato

- Usa l'endpoint primario e l'endpoint di lettura anziché i nomi DNS per nodo nel codice dell'applicazione. Quando il failover automatico è disabilitato, il ridimensionamento, l'applicazione di patch, il failover e altre procedure gestite automaticamente dal servizio ElastiCache quando il failover automatico è abilitato vengono invece eseguite dall'utente. Ciò consente di tenere traccia dei diversi endpoint manualmente. Poiché i vecchi nomi DNS per nodo vengono eliminati e nuovi vengono generati durante la migrazione del cluster da no-TLS a TLS-preferred, non utilizzare direttamente i nomi DNS per nodo. Questo è obbligatorio per consentire ai client di connettersi al cluster durante la migrazione TLS. Inoltre, è possibile distribuire uniformemente le letture tra le repliche quando si utilizza l'endpoint di lettura e tenere traccia dei record DNS quando si aggiungono o si eliminano repliche dal cluster.
- Avere l'endpoint di configurazione del cluster codificato nell'applicazione è una bad practice poiché può essere modificato durante il processo di migrazione TLS.

Durante la crittografia dei dati in transito: prestare attenzione a quando il processo di migrazione termina

La modifica della modalità di crittografia dei dati in transito non è immediata e può richiedere tempo. Ciò vale soprattutto per cluster di grandi dimensioni. Solo al termine della migrazione del cluster a TLS-Preferred sarà in grado di accettare e servire connessioni TCP e TLS. Pertanto, si consiglia di

non creare client che tenteranno di stabilire connessioni TLS al cluster finché la crittografia dei dati in transito non è terminata.

Esistono diversi modi per ricevere una notifica quando la crittografia dei dati in transito viene completata correttamente o non è riuscita: (non mostrato nell'esempio di codice precedente):

- Utilizzo del servizio SNS per ricevere una notifica quando la crittografia è terminata
- Utilizzo dell'API `describe-events` che genera un evento al termine della crittografia
- Visualizzazione di un messaggio nella console ElastiCache che indica che la crittografia è terminata

Puoi anche implementare logica nell'applicazione per sapere se la crittografia è terminata.

Nell'esempio precedente, sono stati illustrati diversi modi per garantire che la migrazione del cluster venga completata:

- Attendere l'avvio del processo di migrazione (lo stato del cluster diventa "in corso di modifica") e attendere il completamento della modifica (lo stato del cluster torna a "disponibile")
- Affermando che nel cluster `transit_encryption_enabled` è impostato su `True` eseguendo query sull'API `describe-replication-group`.

Dopo l'abilitazione della crittografia dei dati in transito: accertarsi che i client utilizzati siano configurati correttamente

Mentre il cluster è in modalità `TLS-preferred`, l'applicazione deve aprire connessioni TLS al cluster e utilizzare solo tali connessioni. In questo modo, nell'applicazione non si verificheranno tempi di inattività durante l'abilitazione della crittografia dei dati in transito. Puoi verificare che non siano presenti connessioni TCP più trasparenti al motore Redis utilizzando il comando `Redis info` nella sezione `SSL`.

```
# SSL
ssl_enabled:yes
ssl_current_certificate_not_before_date:Mar 20 23:27:07 2017 GMT
ssl_current_certificate_not_after_date:Feb 24 23:27:07 2117 GMT
ssl_current_certificate_serial:D8C7DEA91E684163
tls_mode_connected_tcp_clients:0 (should be zero)
tls_mode_connected_tls_clients:100
```


Crittografia dei dati a riposo in ElastiCache

Per mantenere la sicurezza dei dati, Amazon ElastiCache e Amazon S3 forniscono metodi diversi per limitare l'accesso ai dati nella cache. Per ulteriori informazioni, consulta [Amazon VPC e sicurezza di ElastiCache](#) e [Identity and Access Management per Amazon ElastiCache](#).

La crittografia dei dati a riposo di ElastiCache è una funzionalità che incrementa la sicurezza tramite la crittografia dei dati su disco. È sempre abilitata su una cache serverless. Quando è abilitata, esegue la crittografia degli elementi seguenti:

- Il disco durante la sincronizzazione, operazioni di backup e di swap.
- I backup archiviati in Amazon S3

I dati memorizzati su SSD (Solid State Drive) nei cluster abilitati per il tearing di dati sono sempre crittografati.

ElastiCache offre la crittografia dei dati a riposo predefinita (servizio gestito), nonché la possibilità di utilizzare le chiavi simmetriche AWS KMS gestite dal cliente nel [Sistema AWS di gestione delle chiavi](#). Quando viene eseguito il backup della cache, nelle opzioni di crittografia scegli se utilizzare la chiave di crittografia predefinita o una chiave gestita dal cliente. Per ulteriori informazioni, consulta [Abilitazione della crittografia dei dati inattivi](#).

Note

La crittografia predefinita (gestita dal servizio) è l'unica opzione disponibile nelle regioni GovCloud (Stati Uniti).

Important

L'abilitazione della crittografia dei dati a riposo su un cluster Redis progettato autonomamente esistente comporta l'eliminazione del gruppo di replica esistente dopo aver eseguito il backup e il ripristino sul gruppo di replica.

La crittografia dei dati a riposo può essere abilitata su una cache solo quando viene creata. Poiché la crittografia e la decrittografia dei dati richiede l'elaborazione, l'abilitazione della crittografia dei

dati inattivi può in parte influire sulle prestazioni durante queste operazioni. È opportuno creare un riferimento per i dati con o senza crittografia dei dati inattivi per determinare l'impatto sulle prestazioni per i propri casi d'uso.

Argomenti

- [Condizioni di crittografia dei dati inattivi](#)
- [Utilizzo di chiavi gestite dal cliente da KMS AWS](#)
- [Abilitazione della crittografia dei dati inattivi](#)
- [Vedi anche](#)

Condizioni di crittografia dei dati inattivi

Tieni presente i seguenti vincoli sulla crittografia dei dati inattivi di ElastiCache durante la pianificazione della crittografia dei dati inattivi di ElastiCache:

- La crittografia dei dati inattivi è supportata su gruppi di replica che eseguono Redis versione 3.2.6 (pianificata per fine vita (EOL), consultare [Pianificazione della fine del ciclo di vita delle versioni di Redis](#)), 4.0.10 o versioni successive.
- La crittografia dei dati inattivi è supportata solo per gruppi di replica in esecuzione in un Amazon VPC.
- La crittografia dei dati inattivi è supportata solo per i gruppi di replica in esecuzione sui seguenti tipi di nodi.
 - R6gd, R6g, R5, R4, R3
 - M6g, M5, M4, M3
 - T4g, T3, T2

Per ulteriori informazioni, consulta [Tipi di nodi supportati](#)

- La crittografia dei dati inattivi è abilitata impostando in maniera esplicita impostando il parametro `AtRestEncryptionEnabled` su `true`.
- Puoi abilitare la crittografia dei dati inattivi su un gruppo di replica solo durante la creazione del gruppo di replica. Non puoi attivare/disattivare la crittografia dei dati inattivi modificando un gruppo di replica. Per informazioni sull'implementazione della crittografia dei dati inattivi su un gruppo di replica esistente, consulta [Abilitazione della crittografia dei dati inattivi](#).
- Se un cluster utilizza un tipo di nodo della famiglia r6gd, i dati archiviati su SSD vengono crittografati indipendentemente dal fatto che la crittografia a riposo sia abilitata o meno.

- Inoltre, l'opzione di utilizzare le chiavi gestite dal cliente per la crittografia dei dati inattivi non è disponibile nelle regioni AWS GovCloud (us-gov-east-1 e us-gov-west-1).
- Se un cluster utilizza un tipo di nodo della famiglia r6gd, i dati archiviati su SSD vengono crittografati con la chiave AWS KMS gestita dal cliente (o la crittografia gestita dal servizio in Regioni AWS GovCloud) scelta.

L'implementazione della crittografia dei dati inattivi può ridurre le prestazioni durante le operazioni di backup e sincronizzazione. Raffronta la crittografia dei dati inattivi con l'assenza di crittografia sui dati per determinare il suo impatto sulle prestazioni per l'implementazione.

Utilizzo di chiavi gestite dal cliente da KMS AWS

ElastiCache supporta le chiavi simmetriche AWS KMS (chiavi KMS) gestite dal cliente per la crittografia di dati a riposo. 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 [AWSChiavi KMS](#) nella AWS Key Management Service Developer Guide (Guida Sviluppatore del Servizio di Gestione Chiave). Le chiavi devono essere create in AWS KMS prima di poter essere utilizzate in ElastiCache.

Per informazioni su come creare chiavi master KMS AWS, consulta [Creazione di chiavi](#) nella AWSGuida per gli sviluppatori di Key Management Service.

ElastiCache consente l'integrazione con AWS KMS. Per ulteriori informazioni, consulta [Utilizzo di concessioni](#) nella AWSGuida per gli sviluppatori Key Management Service. Non è necessaria nessuna operazione per abilitare l'integrazione Amazon ElastiCache con KMSAWS.

La chiave di condizione `kms:ViaService` limita l'uso di unAWS Chiave KMS (chiave KMS) alle richieste da servizi AWS specificati. Per utilizzare `kms:ViaService` con ElastiCache, includere entrambi i nomi di ViaService nella chiave valore di condizione: `elasticache.AWS_region.amazonaws.com` e `dax.AWS_region.amazonaws.com`. Per ulteriori informazioni, consulta [kms:ViaService](#).

È possibile utilizzare [CloudTrailAWS](#) per monitorare le richieste che Amazon ElastiCache invia a AWS Key Management Service per tuo conto. Tutte le chiamate API a AWS Key Management Service riguardo le chiavi gestite dal cliente hanno registri CloudTrail corrispondenti. Puoi anche vedere le concessioni che ElastiCache crea chiamando l'API KMS [ListGrants](#).

Dopo che un gruppo di replica viene crittografato mediante le chiavi gestite dal cliente, tutti i backup del gruppo di replica sono crittografati nel modo seguente:

- I backup automatici giornalieri vengono crittografati utilizzando la chiave gestita dal cliente associata al cluster.
- Anche il backup finale creato al momento dell'eliminazione del gruppo di replica viene crittografato utilizzando la chiave gestita da cliente associata a gruppo di replica.
- I backup creati manualmente vengono crittografati per impostazione predefinita per utilizzare la chiave KMS associata al gruppo di replica. Puoi sostituirla scegliendo un'altra chiave gestita dal cliente.
- La copia di un backup viene impostata in modo di default sull'uso della chiave gestita dal cliente associata a backup fonte. Puoi sostituirla scegliendo un'altra CMK gestita dal cliente.

Note

- Le chiavi gestite dal cliente non possono essere utilizzate durante l'esportazione dei backup sul bucket Amazon S3 selezionato. Tuttavia, tutti i backup esportati in Amazon S3 vengono crittografati utilizzando la [crittografia lato server](#). Puoi scegliere di copiare i file di backup 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 i backup creati manualmente per i gruppi di replica che, per la crittografia, non utilizzano chiavi gestite dal cliente. Con questa opzione, il file di backup archiviato in Amazon S3 viene crittografato utilizzando una chiave KMS, anche se i dati non sono crittografati sul gruppo di replica di fonte.

Il ripristino da un backup consente di scegliere tra le opzioni di crittografia disponibili, come si fa con le opzioni di crittografia disponibili quando si crea un nuovo gruppo di replica.

- Se elimini o [disabiliti](#) la chiave e [revochi le assegnazioni](#) della chiave utilizzata per crittografare un gruppo di replica, la cache 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. Una volta eliminata la chiave, puoi utilizzare una chiave gestita dal cliente differente per creare un backup per scopi di archiviazione.

- La rotazione automatica delle chiavi permette di mantenere le proprietà delle chiavi root KMS AWS e di conseguenza non ha effetto sulla tua capacità di accedere ai dati ElastiCache. Le cache Amazon ElastiCache crittografate 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 [RotazioneAWS delle chiavi KMS](#) nella AWSGuida per gli sviluppatori di Key Management Service.
- La crittografia di una cache ElastiCache mediante una chiave KMS richiede una singola assegnazione per ogni cache. Questa assegnazione viene utilizzata per tutta la durata della cache. Inoltre, durante la creazione del backup viene utilizzata una singola assegnazione per ogni backup. Questa assegnazione viene ritirata una volta creato il backup.
- Per ulteriori informazioni su concessioni o limiti KMSAWS, consulta [Limiti](#) nella AWSGuida per gli sviluppatori di Key Management Service.

Abilitazione della crittografia dei dati inattivi

Tutte le cache serverless dispongono della crittografia dei dati a riposo abilitata.

Quando si crea un cluster progettato autonomamente, è possibile abilitare la crittografia dei dati a riposo impostando il parametro `AtRestEncryptionEnabled` su `true`. Non puoi abilitare la crittografia dei dati inattivi su gruppi di replica esistenti.

Puoi abilitare la crittografia dei dati a riposo solo durante la creazione di una cache ElastiCache. A tale scopo, è possibile utilizzare il comandoAWS Management Console, ilAWS CLI o l'API ElastiCache.

Durante la creazione di una cache, puoi scegliere una delle opzioni seguenti:

- Predefinita – Questa opzione utilizza la crittografia dei dati inattivi gestita dal servizio.
- Chiave gestita dal cliente – Questa opzione consente di fornire l'ID/ARN della chiave da KMS AWS per la crittografia dei dati inattivi.

Per informazioni su come creare chiavi root KMSAWS, consulta [Creare chiavi](#) nella AWSGuida per gli sviluppatori di Key Management Service

Indice

- [Abilitazione della crittografia dei dati inattivi mediante la AWS Management Console](#)
- [Abilitazione della crittografia dei dati inattivi mediante la AWS CLI](#)

Abilitazione della crittografia dei dati a riposo su un cluster Redis progettato autonomamente esistente

Puoi abilitare la crittografia dei dati inattivi solo durante la creazione di un gruppo di replica Redis. Se disponi di un gruppo di replica esistente su cui desideri abilitare la crittografia dei dati inattivi, procedi nel modo seguente.

Per abilitare la crittografia dei dati inattivi su un gruppo di replica esistente

1. Creare un backup manuale del gruppo di replica esistente. Per ulteriori informazioni, consulta [Esecuzione di backup manuali](#).
2. Creare un nuovo gruppo di replica eseguendo il ripristino dal backup. Sul nuovo gruppo di replica, abilitare la crittografia dei dati inattivi. Per ulteriori informazioni, consulta [Ripristino da un backup in una nuova cache](#).
3. Aggiornare gli endpoint nell'applicazione affinché facciano riferimento al nuovo gruppo di replica.
4. Eliminare il vecchio gruppo di replica. Per ulteriori informazioni, consulta [Eliminazione di un cluster](#) o [Eliminazione di un gruppo di replica](#).

Abilitazione della crittografia dei dati inattivi mediante la AWS Management Console

Abilitazione della crittografia dei dati a riposo su una cache serverless (console)

Tutte le cache serverless dispongono della crittografia dei dati a riposo abilitata. La chiave KMS di proprietà di AWS viene utilizzata per impostazione predefinita per crittografare i dati. Per scegliere la tua chiave AWS KMS, procedi come segue:

- Espandi la sezione Impostazioni predefinite.
- Scegli Personalizza le impostazioni predefinite nella sezione Impostazioni predefinite.
- Scegli Personalizza le impostazioni di sicurezza nella sezione Sicurezza.
- Scegli CMK gestita dal cliente per l'impostazione Chiave di crittografia.
- Seleziona una chiave nell'impostazione Chiave AWS KMS.

Abilitazione della crittografia dei dati a riposo su un cluster progettato autonomamente (console)

Quando si progetta la propria cache, le configurazioni "Sviluppo/Test" e "Produzione" con il metodo "Crea semplice" hanno la crittografia dei dati a riposo abilitata con la chiave predefinita. Quando scegli la configurazione, procedi come segue:

- Scegliere la versione 3.2.6, 4.0.10 o successive come versione motore.
- Fai clic sulla casella di controllo Abilita accanto all'opzione Crittografia dei dati a riposo.
- Scegli una chiave predefinita o una CMK gestita dal cliente.

Per la procedura dettagliata, consulta la documentazione seguente:

- [Creazione di un cluster Redis \(modalità cluster disabilitata\) \(Console\)](#)
- [Creazione di un cluster Redis \(modalità cluster abilitata\) \(console\)](#).

Abilitazione della crittografia dei dati inattivi mediante la AWS CLI

Per abilitare la crittografia dei dati inattivi durante la creazione di un cluster Redis mediante AWS CLI, utilizza il parametro `--at-rest-encryption-enabled` durante la creazione di un gruppo di replica.

Abilitazione della crittografia dei dati inattivi su un cluster Redis (CLI) (modalità cluster disabilitata)

La seguente operazione consente di creare il gruppo di replica replica Redis (modalità cluster disattivata) `my-classic-rg` con tre nodi (`--num-cache-clusters`), un nodo primario e due nodi di replica di lettura. La crittografia dei dati inattivi è abilitata per questo gruppo di replica (`--at-rest-encryption-enabled`).

I seguenti parametri e i relativi valori sono necessari per abilitare la crittografia su questo gruppo di replica:

Parametri chiave

- `--engine` — Deve essere `redis`.
- `--engine-version` - Deve essere 3.2.6, 4.0.10 o versione successiva.
- `--at-rest-encryption-enabled`- Richiesto per abilitare la crittografia inattiva.

Example 1: Redis (modalità cluster disattivata) con repliche

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id my-classic-rg \  
  --replication-group-description "3 node replication group" \  
  --at-rest-encryption-enabled
```

```
--cache-node-type cache.m4.large \  
--engine redis \  
--at-rest-encryption-enabled \  
--num-cache-clusters 3
```

Per Windows:

```
aws elasticache create-replication-group ^  
--replication-group-id my-classic-rg ^  
--replication-group-description "3 node replication group" ^  
--cache-node-type cache.m4.large ^  
--engine redis ^  
--at-rest-encryption-enabled ^  
--num-cache-clusters 3 ^
```

Per ulteriori informazioni, consulta la seguente documentazione:

- [Creazione da zero di un gruppo di replica Redis \(modalità cluster disabilitata\) \(AWS CLI\)](#)
- [create-replication-group](#)

Abilitazione della crittografia dei dati inattivi su un cluster per Redis (modalità cluster attivata) (CLI)

La seguente operazione consente di creare il gruppo di replica Redis (modalità cluster abilitata) `my-clustered-rg` con tre gruppi di nodi o partizioni (`--num-node-groups`). Ciascuno dispone di tre nodi, un nodo primario e due nodi replica di lettura (`--replicas-per-node-group`). La crittografia dei dati inattivi è abilitata per questo gruppo di replica (`--at-rest-encryption-enabled`).

I seguenti parametri e i relativi valori sono necessari per abilitare la crittografia su questo gruppo di replica:

Parametri chiave

- **--engine** — Deve essere `redis`.
- **--engine-version**—Deve essere `4.0.10` o versione successiva.
- **--at-rest-encryption-enabled**- Richiesto per abilitare la crittografia inattiva.
- **--cache-parameter-group**: deve essere `default-redis4.0.cluster.on` o uno derivato da esso per rendere questo un gruppo di replica abilitato in modalità cluster.

Example 2: Un cluster Redis (modalità cluster abilitata)

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id my-clustered-rg \  
  --replication-group-description "redis clustered cluster" \  
  --cache-node-type cache.m3.large \  
  --num-node-groups 3 \  
  --replicas-per-node-group 2 \  
  --engine redis \  
  --engine-version 6.2 \  
  --at-rest-encryption-enabled \  
  --cache-parameter-group default.redis6.x.cluster.on
```

Per Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id my-clustered-rg ^  
  --replication-group-description "redis clustered cluster" ^  
  --cache-node-type cache.m3.large ^  
  --num-node-groups 3 ^  
  --replicas-per-node-group 2 ^  
  --engine redis ^  
  --engine-version 6.2 ^  
  --at-rest-encryption-enabled ^  
  --cache-parameter-group default.redis6.x.cluster.on
```

Per ulteriori informazioni, consulta la seguente documentazione:

- [Creazione da zero di un gruppo di replica Redis \(modalità cluster abilitata\) \(AWS CLI\)](#)
- [create-replication-group](#)

Vedi anche

- [Amazon VPC e sicurezza di ElastiCache](#)
- [Identity and Access Management per Amazon ElastiCache](#)

Autenticazione e autorizzazione

ElastiCache supporta l'autenticazione degli utenti mediante IAM e il comando AUTH Redis e l'autorizzazione delle operazioni degli utenti mediante il controllo degli accessi basato su ruolo (RBAC).

Argomenti

- [Controllo accessi basato sui ruoli \(RBAC\)](#)
- [Autenticazione con il comando Redis AUTH](#)
- [Disabilitazione del controllo degli accessi su una cache ElastiCache Redis](#)

Controllo accessi basato sui ruoli (RBAC)

Invece di autenticare gli utenti con il comando Redis AUTH come descritto in [Autenticazione con il comando Redis AUTH](#), in Redis 6.0 e successive è possibile utilizzare una caratteristica denominata Controllo degli accessi basati su ruoli (RBAC). Il controllo degli accessi basato su ruoli è anche l'unico modo per controllare l'accesso alle cache serverless.

A differenza di Redis AUTH, in cui tutti i client autenticati dispongono dell'accesso completo alla cache se il token è autenticato, il controllo degli accessi basato su ruoli consente di controllare l'accesso alla cache tramite i gruppi di utenti. Questi gruppi di utenti sono progettati come un modo per organizzare l'accesso alle cache.

Con RBAC, è possibile creare utenti e assegnare loro autorizzazioni specifiche utilizzando una stringa di accesso, come descritto di seguito. Assegna gli utenti ai gruppi di utenti allineati con un ruolo specifico (amministratori, risorse umane) che vengono poi implementati in una o più cache ElastiCache per Redis. In questo modo, è possibile stabilire limiti di sicurezza tra i client utilizzando la stessa o le stesse cache Redis e impedire ai client di accedere ai dati reciproci.

RBAC è progettato per supportare l'introduzione di [ACL Redis](#) in Redis 6. Quando utilizzi RBAC con la cache ElastiCache per Redis, si applicano le seguenti limitazioni:

- Non è possibile specificare password in una stringa di accesso. È possibile impostare le password con chiamate [CreaUser \(CreaUtente\)](#) o [ModifyUser \(ModificaUtente\)](#).
- Per i diritti utente, si passa `on` come parte della stringa di accesso. Se nessuno dei due è specificato nella stringa di accesso, all'utente viene assegnato `off` e non dispone dei diritti di accesso alla cache.

- Non è possibile utilizzare comandi proibiti e rinominati. Se si specifica un comando proibito o rinominato, verrà generata un'eccezione. Se si desidera utilizzare elenchi di controllo di accesso (ACL) per un comando rinominato, specificare il nome originale del comando, ovvero il nome del comando prima che venisse rinominato.
- Non è possibile utilizzare `lareset` come parte di una stringa di accesso. È possibile specificare le password con i parametri API e ElastiCache for Redis gestisce le password. Pertanto, non è possibile utilizzare `reset` perché rimuoverebbe tutte le password per un utente.
- Redis 6 introduce la [ELENCO ACL](#) comando. Questo comando restituisce un elenco di utenti insieme alle regole ACL applicate a ciascun utente. ElastiCache for Redis supporta `ACL LIST`, ma non include il support per gli hash delle password come fa Redis. Con ElastiCache for Redis, puoi usare il [describe-users](#) per ottenere informazioni simili, incluse le regole contenute nella stringa di accesso. Tuttavia, [describe-users](#) non recupera una password utente.

Altri comandi di sola lettura supportati da ElastiCache for Redis includono [ACL WHOAMI](#), [GLI UTENTI DI ACL](#), e [ACL CAT](#). ElastiCache for Redis non supporta altri comandi ACL basati su scrittura.

- Vengono applicati i vincoli seguenti:

Risorsa	Massimo consentito
Utenti per gruppo di utenti	100
Numero di DPU	1000
Numero di gruppi di utenti	100

Utilizzando RBAC con ElastiCache for Redis è descritto in modo più dettagliato di seguito.

Argomenti

- [Specifica delle autorizzazioni mediante una stringa di accesso](#)
- [Applicazione di RBAC a una cache per ElastiCache per Redis](#)
- [Migrazione da Redis AUTH a RBAC](#)
- [Migrazione da RBAC a Redis AUTH](#)
- [Rotazione automatica delle password per utenti](#)
- [Autenticazione con IAM](#)

Specifica delle autorizzazioni mediante una stringa di accesso

Per specificare le autorizzazioni per una cache ElastiCache per Redis, crea una stringa di accesso e assegnala a un utente, utilizzando la AWS CLI o la 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 da sinistra a destra cumulativamente e una stringa più semplice può essere utilizzata al posto di quella fornita se ci sono ridondanze nella stringa 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.
- `+@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 e gruppi di utenti con Console e CLI](#).

Se si esegue la migrazione di un carico di lavoro esistente a ElastiCache, è possibile recuperare la stringa di accesso chiamando `ACL LIST`, escludendo l'utente e gli hash delle password.

Per Redis versione 6.2 e successive è supportata anche la seguente sintassi della stringa di accesso:

- `&*` – L'accesso è consentito a tutti i canali disponibili.

Per Redis versione 7.0 e successive è supportata anche la seguente sintassi della stringa di accesso:

- `|` – Può essere usato per bloccare i sottocomandi (ad esempio, `"-config|set"`).
- `%R~<pattern>` – Aggiunge il modello di chiave di lettura specificato. Il comportamento è simile al modello chiave normale, ma concede l'autorizzazione solo per leggere dalle chiavi che corrispondono al modello specificato. Per ulteriori informazioni, consulta [autorizzazioni della chiave](#).
- `%W~<pattern>` – Aggiunge il modello di chiave di scrittura specificato. Il comportamento è simile al modello chiave normale, ma concede solo l'autorizzazione per scrivere nelle chiavi che corrispondono al modello specificato. Per ulteriori informazioni, consulta [autorizzazioni della chiave](#).
- `%RW~<pattern>` – Alias per `~<pattern>`.
- `(<rule list>)` – Crea un nuovo selettore rispetto al quale abbinare le regole. I selettori vengono valutati dopo le autorizzazioni utente e in base all'ordine in cui sono definiti. Se un comando corrisponde alle autorizzazioni utente o a qualsiasi selettore, viene consentito. Per ulteriori informazioni, consulta [Selettori ACL](#).
- `clearselectors` – Elimina tutti i selettori collegati all'utente.

Applicazione di RBAC a una cache per ElastiCache per Redis

Per utilizzare ElastiCache for Redis RBAC, procedi nel seguente modo:

1. Crea uno o più utenti.
2. Crea un gruppo di utenti e aggiungi utenti al gruppo.
3. Assegna il gruppo di utenti a una cache in cui è abilitata la crittografia dei dati in transito.

La tabella seguente descrive i seguenti passaggi nel dettaglio.

Argomenti

- [Creazione di utenti e gruppi di utenti con Console e CLI](#)
- [Gestione dei gruppi di utenti con Console e CLI](#)

- [Assegnazione di gruppi di utenti alle cache serverless](#)
- [Assegnazione di gruppi di utenti ai gruppi di replica](#)

Creazione di utenti e gruppi di utenti con Console e CLI

Le informazioni utente per gli utenti RBAC sono un ID utente, 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. L'ID utente è univoco per l'utente e il nome utente è ciò che viene passato al motore.

Assicurarsi che le autorizzazioni utente fornite abbiano senso con lo scopo previsto del gruppo di utenti. Ad esempio, se si crea un gruppo di utenti denominato `Administrators`, qualsiasi utente aggiunto a quel gruppo deve avere la relativa stringa di accesso impostata per l'accesso completo a tasti e comandi. Per gli utenti in `une-commerce`, è possibile impostare le relative stringhe di accesso su accesso di sola lettura.

ElastiCache configura automaticamente un utente di default con ID utente e nome utente `default` e lo aggiunge a tutti i gruppi di utenti. 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.

Per aggiungere a una cache un controllo di accesso appropriato, sostituisci l'utente predefinito con uno nuovo che non è abilitato o utilizza una password complessa. Per modificare l'utente di default, creare un nuovo utente con il nome utente impostato su `default`. È quindi possibile scambiarlo con l'utente di default originale.

Di seguito viene illustrata la procedura per scambiare l'utente originale `default` con un altro `default` utente che dispone di una stringa di accesso modificata.

Come modificare l'utente predefinito nella console

1. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione seleziona Gestione gruppi di utenti.
3. Per ID gruppo di utenti scegli l'ID che desideri modificare. Verificare di aver selezionato il collegamento e non la casella di controllo.
4. Scegli Modifica.

5. Nella finestra Modifica scegli Gestisci e per Nome utente seleziona l'utente che desideri per impostare come predefinito.
6. Scegliere Choose (Scegli).
7. Scegli Modifica. Quando si esegue questa operazione vengono terminate tutte le connessioni esistenti a una cache disponibili per l'utente predefinito originale.

Per modificare l'utente predefinito tramite AWS CLI

1. Crea un nuovo utente con il nome `default` utilizzando i seguenti comandi.

Per Linux, macOS o Unix:

```
aws elasticache create-user \  
  --user-id "new-default-user" \  
  --user-name "default" \  
  --engine "REDIS" \  
  --passwords "a-strong-password" \  
  --access-string "off +get ~keys*"
```

Per Windows:

```
aws elasticache create-user ^  
  --user-id "new-default-user" ^  
  --user-name "default" ^  
  --engine "REDIS" ^  
  --passwords "a-strong-password" ^  
  --access-string "off +get ~keys*"
```

2. Creare un gruppo di utenti e aggiungere l'utente creato in precedenza.

Per Linux, macOS o Unix:

```
aws elasticache create-user-group \  
  --user-group-id "new-group-2" \  
  --engine "REDIS" \  
  --user-ids "new-default-user"
```

Per Windows:

```
aws elasticache create-user-group ^
```

```
--user-group-id "new-group-2" ^  
--engine "REDIS" ^  
--user-ids "new-default-user"
```

3. Scambia il nuovo default utente con l'originale default utente.

Per Linux, macOS o Unix:

```
aws elasticache modify-user-group \  
  --user-group-id test-group \  
  --user-ids-to-add "new-default-user" \  
  --user-ids-to-remove "default"
```

Per Windows:

```
aws elasticache modify-user-group ^  
  --user-group-id test-group ^  
  --user-ids-to-add "new-default-user" ^  
  --user-ids-to-remove "default"
```

Quando viene chiamata questa operazione di modifica, vengono terminate tutte le connessioni esistenti a una cache disponibili per l'utente predefinito originale.

Quando si crea un utente, è possibile impostare fino a due password. Quando si modifica una password, vengono mantenute tutte le connessioni esistenti alle cache.

In particolare, considera le seguenti limitazioni della password dell'utente quando utilizzi RBAC per ElastiCache for Redis:

- 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

Utilizza la procedura seguente per gestire gli utenti sulla console.

Per gestire gli utenti nella console

1. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.

2. Sul pannello di controllo Amazon ElastiCache seleziona Gestione utente. Sono disponibili le seguenti opzioni:

- **Crea utente:** quando si crea un utente, si immette un ID utente, un nome utente, una modalità di autenticazione e una stringa di accesso. La stringa di accesso imposta il livello di autorizzazione per le chiavi e i comandi consentiti all'utente.

Quando si crea un utente, è possibile impostare fino a due password. Quando si modifica una password, vengono mantenute tutte le connessioni esistenti alle cache.

- **Modifica utente:** consente di aggiornare le impostazioni di autenticazione di un utente o di modificarne la stringa di accesso.
- **Elimina utente:** l'account viene rimosso da qualsiasi gruppo di utenti a cui appartiene.

Utilizza le procedure seguenti per gestire gli utenti con l'AWS CLI.

Per modificare un utente utilizzando la CLI;

- Utilizzo dell'`modify-user` per aggiornare la password o le password di un utente o modificare le autorizzazioni di accesso di un utente.

Quando un utente viene modificato, vengono aggiornati i gruppi di utenti associati all'utente, insieme a tutte le cache associate ai gruppi di utenti. Tutte le connessioni esistenti vengono mantenute. Di seguito vengono mostrati gli esempi.

Per Linux, macOS o Unix:

```
aws elasticache modify-user \  
  --user-id user-id-1 \  
  --access-string "~objects:* ~items:* ~public:*" \  
  --no-password-required
```

Per Windows:

```
aws elasticache modify-user ^  
  --user-id user-id-1 ^  
  --access-string "~objects:* ~items:* ~public:*" ^  
  --no-password-required
```

Note

Non è consigliabile utilizzare l'opzione `lanopass`. In tal caso, si consiglia di impostare le autorizzazioni dell'utente in sola lettura con accesso a un set limitato di chiavi.

Per eliminare un utente utilizzando la CLI;

- Utilizza il comando `delete-user` per eliminare un utente. L'account viene eliminato e rimosso da tutti i gruppi di utenti a cui appartiene. Di seguito è riportato un esempio.

Per Linux, macOS o Unix:

```
aws elasticache delete-user \  
  --user-id user-id-2
```

Per Windows:

```
aws elasticache delete-user ^  
  --user-id user-id-2
```

Per visualizzare un elenco di utenti, chiama l'operazione [describe-users](#).

```
aws elasticache describe-users
```

Gestione dei gruppi di utenti con Console e CLI

È possibile creare gruppi di utenti per organizzare e controllare l'accesso degli utenti a una o più cache, come illustrato di seguito.


Utilizza la procedura seguente per gestire gruppi di utenti utilizzando la console.

Gestione dei gruppi di utenti tramite la console;

1. Accedi alla AWS Management Console e apri la console di Amazon ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Sul pannello di controllo Amazon ElastiCache seleziona Gestione gruppi di utenti.

Di seguito sono elencate le operazioni disponibili per creare nuovi gruppi di utenti:

- **Crea:** quando si crea un gruppo di utenti, si aggiungono gli utenti e quindi si assegnano i gruppi di utenti alle cache. Ad esempio, è possibile creare un gruppo di utenti Admin per gli utenti che dispongono di ruoli amministrativi per una cache.


 Important

Quando si crea un gruppo di utenti, è necessario includere l'utente predefinito.

- **Aggiungi utenti:** aggiunge utenti al gruppo di utenti.
- **Rimuovere gli utenti—** Rimuove gli utenti dal gruppo di utenti. Quando gli utenti vengono rimossi da un gruppo di utenti, vengono terminate tutte le connessioni esistenti alle cache.
- **Elimina:** utilizza questa opzione per eliminare un gruppo di utenti. Si noti che il gruppo di utenti stesso, non gli utenti che appartengono al gruppo, verrà eliminato.

Per i gruppi di utenti esistenti, puoi effettuare le seguenti operazioni:

- **Aggiungi utenti—** Aggiunge utenti esistenti al gruppo di utenti.
- **Eliminazione di utenti—** Rimuove gli utenti esistenti dal gruppo di utenti.

 Note

Gli utenti vengono rimossi dal gruppo di utenti, ma non eliminati dal sistema.

Utilizza le procedure seguenti per gestire gruppi di utenti utilizzando la CLI.

Per creare un nuovo gruppo di utenti e aggiungere un utente utilizzando la CLI

- Usa il comando `create-user-group` come mostrato nell'esempio seguente.

Per Linux, macOS o Unix:

```
aws elasticache create-user-group \  
  --user-group-id "new-group-1" \  
  --engine "REDIS" \  
  --user-ids user-id-1, user-id-2
```

Per Windows:

```
aws elasticache create-user-group ^
  --user-group-id "new-group-1" ^
  --engine "REDIS" ^
  --user-ids user-id-1, user-id-2
```

Per modificare un gruppo di utenti aggiungendo nuovi utenti o rimuovendo membri correnti tramite la CLI

- Usa il comando `modify-user-group` come mostrato nell'esempio seguente.

Per Linux, macOS o Unix:

```
aws elasticache modify-user-group --user-group-id new-group-1 \
  --user-ids-to-add user-id-3 \
  --user-ids-to-remove user-id-2
```

Per Windows:

```
aws elasticache modify-user-group --user-group-id new-group-1 ^
  --user-ids-to-add userid-3 ^
  --user-ids-to-remove user-id-2
```

Note

Tutte le connessioni aperte appartenenti a un utente rimosse da un gruppo di utenti vengono terminate con questo comando.

Per eliminare un gruppo di utenti tramite la CLI;

- Usa il comando `delete-user-group` come mostrato nell'esempio seguente. Il gruppo di utenti stesso, non gli utenti che appartengono al gruppo, viene eliminato.

Per Linux, macOS o Unix:

```
aws elasticache delete-user-group /
  --user-group-id
```

Per Windows:

```
aws elasticache delete-user-group ^  
  --user-group-id
```

Per visualizzare un elenco di gruppi di utenti, è possibile chiamare il metodo [describe-user-groups](#) operazione.

```
aws elasticache describe-user-groups \  
  --user-group-id test-group
```

Assegnazione di gruppi di utenti alle cache serverless

Dopo aver creato un gruppo di utenti e aggiunto gli utenti, il passaggio finale dell'implementazione di RBAC consiste nell'assegnare il gruppo di utenti a una cache serverless.

Assegnazione di gruppi di utenti alle cache serverless mediante la console

Per aggiungere un gruppo di utenti a una cache serverless mediante la AWS Management Console, effettua le seguenti operazioni:

- Per la modalità cluster disabilitata, vedere [Creazione di un cluster Redis \(modalità cluster disabilitata\) \(Console\)](#)
- Per la modalità cluster abilitata, consulta [Creazione di un cluster Redis \(modalità cluster abilitata\) \(console\)](#).

Assegnazione di gruppi di utenti alle cache serverless mediante la AWS CLI

La seguente operazione della AWS CLI crea una cache serverless utilizzando il parametro `user-group-id` con il valore *my-user-group-id*. Sostituisci il gruppo di sottoreti `sng-test` con uno esistente.

Parametri chiave

- **--engine** — Deve essere `redis`.
- **--user-group-id**: questo valore fornisce l'ID del gruppo di utenti composto da utenti con autorizzazioni di accesso specificate per la cache.

Per Linux, macOS o Unix:

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name "new-serverless-cache" \  
  --description "new-serverless-cache" \  
  --engine "redis" \  
  --user-group-id "new-group-1"
```

Per Windows:

```
aws elasticache create-serverless-cache ^  
  --serverless-cache-name "new-serverless-cache" ^  
  --description "new-serverless-cache" ^  
  --engine "redis" ^  
  --user-group-id "new-group-1"
```

L'operazione AWS CLI seguente modifica una cache serverless con il parametro `user-group-id` impostato sul valore *my-user-group-id*.

Per Linux, macOS o Unix:

```
aws elasticache modify-serverless-cache \  
  --serverless-cache-name serverless-cache-1 \  
  --user-group-id "new-group-2"
```

Per Windows:

```
aws elasticache modify-serverless-cache ^  
  --serverless-cache-name serverless-cache-1 ^  
  --user-group-id "new-group-2"
```

Tieni presente che tutte le modifiche apportate a una cache vengono aggiornate in modo asincrono. Puoi monitorare l'avanzamento visualizzando gli eventi. Per ulteriori informazioni, consulta [Visualizzazione di eventi ElastiCache C](#).

Assegnazione di gruppi di utenti ai gruppi di replica

Dopo aver creato un gruppo di utenti e aggiunto utenti, il passaggio finale nell'implementazione di RBAC consiste nell'assegnare il gruppo di utenti a un gruppo di replica.

Assegnazione di gruppi di utenti ai gruppi di replica mediante la console

Per aggiungere un gruppo di utenti a una replica mediante il comando AWS Management Console Effettua le seguenti operazioni:

- Per la modalità cluster disabilitata, vedere [Creazione di un cluster Redis \(modalità cluster disabilitata\) \(Console\)](#)
- Per la modalità cluster abilitata, consulta [Creazione di un cluster Redis \(modalità cluster abilitata\) \(console\)](#).

Assegnazione di gruppi di utenti ai gruppi di replica mediante il comando AWS CLI

La seguente operazione AWS CLI crea un gruppo di replica con crittografia in transito (TLS) abilitata e il parametro `user-group-ids` con il valore `my-user-group-id`. Sostituisci il gruppo di sottoreti `sng-test` con uno esistente.

Parametri chiave

- **--engine** — Deve essere `redis`.
- **--engine-version** – Deve essere versione 6.0 o successiva.
- **--transit-encryption-enabled**— Richiesto per l'autenticazione e per l'associazione di un gruppo di utenti.
- **--user-group-ids**: questo valore fornisce l'ID del gruppo di utenti composto da utenti con autorizzazioni di accesso specificate per la cache.
- **--cache-subnet-group**: obbligatorio per l'associazione di un gruppo di utenti.

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id "new-replication-group" \  
  --replication-group-description "new-replication-group" \  
  --engine "redis" \  
  --cache-node-type cache.m5.large \  
  --transit-encryption-enabled \  
  --user-group-ids "new-group-1" \  
  --cache-subnet-group "cache-subnet-group"
```

Per Windows:

```
aws elasticache create-replication-group ^
  --replication-group-id "new-replication-group" ^
  --replication-group-description "new-replication-group" ^
  --engine "redis" ^
  --cache-node-type cache.m5.large ^
  --transit-encryption-enabled ^
  --user-group-ids "new-group-1" ^
  --cache-subnet-group "cache-subnet-group"
```

La seguente operazione AWS CLI modifica un gruppo di replica con crittografia in transito (TLS) abilitata e il parametro `user-group-ids` con il valore `my-user-group-id`.

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
  --replication-group-id replication-group-1 \
  --user-group-ids-to-remove "new-group-1" \
  --user-group-ids-to-add "new-group-2"
```

Per Windows:

```
aws elasticache modify-replication-group ^
  --replication-group-id replication-group-1 ^
  --user-group-ids-to-remove "new-group-1" ^
  --user-group-ids-to-add "new-group-2"
```

Prendere nota dell'ID `PendingChanges` nella risposta. Tutte le modifiche apportate a una cache vengono aggiornate in modo asincrono. Puoi monitorare l'avanzamento visualizzando gli eventi. Per ulteriori informazioni, consulta [Visualizzazione di eventi ElastiCache C](#).

Migrazione da Redis AUTH a RBAC

Se si utilizza Redis AUTH come descritto in [Autenticazione con il comando Redis AUTH](#) e si desidera eseguire la migrazione all'utilizzo di RBAC, utilizzare le seguenti procedure.

Utilizza la procedura seguente per eseguire la migrazione da Redis AUTH a RBAC utilizzando la console.

Per eseguire la migrazione da Redis AUTH a RBAC utilizzando la console

1. Accedi alla AWS Management Console e apri la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dall'elenco nell'angolo in alto a destra, scegli la regione AWS in cui si trova la cache da modificare.
3. Nel riquadro di navigazione scegli il motore in esecuzione sulla cache da modificare.

Viene visualizzato un elenco di cache che eseguono il motore scelto.

4. Nell'elenco delle cache, scegli il nome della cache che desideri modificare.
5. Per Operazioni, scegli Modifica.

Viene visualizzata la finestra Modifica.

6. Per Controllo accessi scegli Lista di controllo accessi del gruppo di utenti.
7. Per Lista di controllo accessi del gruppo di utenti scegli un gruppo di utenti.
8. Scegli Visualizza l'anteprima delle modifiche e quindi, nella schermata successiva, seleziona Modifica.

Utilizza la procedura seguente per eseguire la migrazione da Redis AUTH a RBAC utilizzando la CLI.

Per eseguire la migrazione da Redis AUTH a RBAC utilizzando l'interfaccia CLI

- Usa il comando `modify-replication-group` come mostrato di seguito.

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group --replication-group-id test \  
  --auth-token-update-strategy DELETE \  
  --user-group-ids-to-add user-group-1
```

Per Windows:

```
aws elasticache modify-replication-group --replication-group-id test ^  
  --auth-token-update-strategy DELETE ^  
  --user-group-ids-to-add user-group-1
```

Migrazione da RBAC a Redis AUTH

Se si utilizza RBAC e si desidera eseguire la migrazione a Redis AUTH, vedere [Migrazione da RBAC a Redis AUTH](#).

Note

Se devi disabilitare il controllo degli accessi per una cache ElastiCache, dovrai usare la AWS CLI. Per ulteriori informazioni, consulta [the section called “Disabilitazione del controllo degli accessi su una cache ElastiCache Redis”](#).

Rotazione automatica delle password per utenti

Con AWS Secrets Manager, puoi sostituire le credenziali nel codice (incluse le password) con una chiamata API a Secrets Manager in modo da recuperare il segreto a livello di codice. Questo approccio aiuta a garantire che il segreto non venga compromesso da qualcuno che esamina il codice, perché semplicemente il segreto non è presente. Inoltre, puoi configurare Secrets Manager affinché ruoti automaticamente il segreto in base a una pianificazione specificata. In questo modo puoi sostituire i segreti a lungo termine con altri a breve termine, contribuendo a ridurre notevolmente il rischio di compromissione.

Utilizzando Secrets Manager, è possibile ruotare automaticamente le password ElastiCache per Redis (ovvero, segreti) utilizzando una funzione AWS Lambda fornita da Secrets Manager.

Per ulteriori informazioni su AWS Secrets Manager, consulta [Che cos'è AWS Secrets Manager](#).

Modalità di utilizzo dei segreti in ElastiCache

Con Redis 6, ElastiCache per Redis ha introdotto [Controllo accessi basato sui ruoli \(RBAC\)](#) per proteggere il cluster Redis. Questa funzionalità consente di limitare determinate connessioni in termini di comandi che è possibile eseguire e tasti a cui è possibile accedere. Con RBAC, mentre il cliente crea un utente con password, i valori delle password devono essere immessi manualmente in testo normale e sono visibili per l'operatore.

Con Secrets Manager, le applicazioni recuperano la password da Secrets Manager anziché inserirle manualmente e archivarle nella configurazione dell'applicazione. Per informazioni su come fare, consulta [Modalità di associazione degli utenti al segreto in ElastiCache](#).

L'utilizzo di segreti comporta un costo. Per informazioni sui prezzi, consulta [Prezzi di AWS Secrets Manager](#).

Modalità di associazione degli utenti al segreto in ElastiCache

Secrets Manager manterrà un riferimento per l'utente associato nel campo `SecretString` del segreto. Da parte di ElastiCache non ci sarà alcun riferimento al segreto.

```
{
  "password": "strongpassword",
  "username": "user1",
  "user_arn": "arn:aws:elasticache:us-east-1:xxxxxxxxxxx918:user:user1" //this is the
  bond between the secret and the user
}
```

Funzione di rotazione Lambda

Per abilitare la rotazione automatica delle password di Secrets Manager, creare una funzione Lambda che interagisce con l'API [modify-user](#) per aggiornare le password dell'utente.

Per informazioni sul funzionamento, consulta [Come funzione la rotazione](#).

Note

Per alcuni servizi AWS, per evitare lo scenario confused deputy, AWS consiglia di utilizzare entrambe le chiavi di condizione globali `aws:SourceArn` e `aws:SourceAccount`. Tuttavia, se includi la condizione `aws:SourceArn` nella tua policy della funzione di rotazione, la funzione di rotazione può essere utilizzata solo per ruotare il segreto specificato da tale ARN. Ti consigliamo di includere solo la chiave di contesto `aws:SourceAccount` in modo da poter utilizzare la funzione di rotazione per più segreti.

Per eventuali problemi che si potrebbero verificare, consulta [Risoluzione dei problemi relativi alla rotazione di AWS Secrets Manager](#).

Come creare un utente ElastiCache e associarlo a Secrets Manager

Nei passaggi seguenti viene illustrato come creare un utente e associarlo a Secrets Manager:

1. Creazione di un utente non attivo

Per Linux, macOS o Unix:

```
aws elasticache create-user \
```

```
--user-id user1 \  
--user-name user1 \  
--engine "REDIS" \  
--no-password \ // no authentication is required  
--access-string "*off* +get ~keys*" // this disables the user
```

Per Windows:

```
aws elasticache create-user ^  
--user-id user1 ^  
--user-name user1 ^  
--engine "REDIS" ^  
--no-password ^ // no authentication is required  
--access-string "*off* +get ~keys*" // this disables the user
```

Noterai una risposta simile alla seguente:

```
{  
  "UserId": "user1",  
  "UserName": "user1",  
  "Status": "active",  
  "Engine": "redis",  
  "AccessString": "off ~keys* -@all +get",  
  "UserGroupIds": [],  
  "Authentication": {  
    "Type": "no_password"  
  },  
  "ARN": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1"  
}
```

2. Creazione di un segreto

Per Linux, macOS o Unix:

```
aws secretsmanager create-secret \  
--name production/ec/user1 \  
--secret-string \  
'{  
  "user_arn": "arn:aws:elasticache:us-east-1:123456xxxx:user:user1",  
  "username": "user1"  
}'
```

Per Windows:

```
aws secretsmanager create-secret ^
--name production/ec/user1 ^
--secret-string ^
'{
  "user_arn": "arn:aws:elasticache:us-east-1:123456xxxx:user:user1",
  "username":"user1"
}'
```

Noterai una risposta simile alla seguente:

```
{
  "ARN": "arn:aws:secretsmanager:us-east-1:123456xxxx:secret:production/ec/user1-
eaFois",
  "Name": "production/ec/user1",
  "VersionId": "aae5b963-1e6b-4250-91c6-ebd6c47d0d95"
}
```

3. Configurazione di una funzione Lambda per ruotare la password

- a. Accedi alla AWS Management Console e apri la console Lambda all'indirizzo <https://console.aws.amazon.com/lambda/>
- b. Nel riquadro di navigazione, seleziona Functions (Funzioni), quindi scegli la funzione da creare. Scegli il nome della funzione, non la casella di controllo alla sua sinistra.
- c. Scegli la scheda Configurazione.
- d. In General configuration (Configurazione generale), scegli Edit (Modifica), quindi imposta Timeout su almeno 12 minuti.
- e. Seleziona Salva.
- f. Scegli Environmental variables (Variabili di ambiente), quindi imposta le seguenti opzioni:
 - i. SECRETS_MANAGER_ENDPOINT – <https://secretsmanager.REGION.amazonaws.com>
 - ii. SECRET_ARN – Il nome della risorsa Amazon (ARN) del segreto creato nella Fase 2.
 - iii. USER_NAME – Nome dell'utente ElastiCache,
 - iv. Seleziona Salva.
- g. Seleziona Permissions (Autorizzazioni)

- h. In Execution role (Ruolo di esecuzione), scegli il nome del ruolo della funzione Lambda da visualizzare sulla console IAM.
- i. La funzione Lambda avrà bisogno della seguente autorizzazione per modificare gli utenti e impostare la password:

ElastiCache

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:DescribeUsers",
        "elasticache:ModifyUser"
      ],
      "Resource": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1"
    }
  ]
}
```

Secrets Manager

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:PutSecretValue",
        "secretsmanager:UpdateSecretVersionStage"
      ],
      "Resource": "arn:aws:secretsmanager:us-east-1:xxxxxxxxxx:secret:XXXX"
    },
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetRandomPassword",
      "Resource": "*"
    }
  ]
}
```

```
]
}
```

4. Impostazione della rotazione segreto di Secrets Manager

- a. Usando la AWS Management Console, consulta [Configurazione della rotazione automatica dei segreti di AWS Secrets Manager utilizzando la console](#)

Per ulteriori informazioni sulla configurazione di una pianificazione della rotazione, consulta la pagina relativa alle [espressioni di pianificazione nella rotazione Secrets Manager](#).

- b. Utilizzando la AWS CLI, consulta [Configurazione della rotazione automatica per AWS Secrets Manager utilizzando AWS Command Line Interface](#)

Autenticazione con IAM

Argomenti

- [Panoramica](#)
- [Limitazioni](#)
- [Installazione](#)
- [Connessione](#)

Panoramica

Con l'autenticazione IAM puoi autenticare una connessione a ElastiCache per Redis utilizzando le identità AWS IAM, quando la cache è configurata per utilizzare Redis 7 o versioni successive. Ciò consente di consolidare il modello di sicurezza e semplificare molte attività di sicurezza amministrative. Con l'autenticazione IAM puoi anche configurare il controllo granulare degli accessi per ogni singola cache ElastiCache e utente ElastiCache e seguire il principio di autorizzazioni del privilegio minimo. L'autenticazione IAM in ElastiCache per Redis funziona fornendo un token di autenticazione IAM di breve durata anziché una password utente ElastiCache di lunga durata nel comando Redis AUTH o 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 assegnare l'accesso agli utenti dai provider di identità federate direttamente alle cache Redis.

Per utilizzare AWS IAM con ElastiCache per Redis, devi prima creare un utente ElastiCache con la modalità di autenticazione impostata su IAM, quindi puoi creare o riutilizzare un'identità IAM. L'identità IAM richiede una policy associata per assegnare l'azione `elasticache:Connect` alla cache e all'utente ElastiCache. 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 alla cache Redis. Un client Redis con supporto per il provider di credenziali può generare automaticamente le credenziali temporanee per ogni nuova connessione. ElastiCache per Redis eseguirà l'autenticazione IAM per le richieste di connessione degli utenti ElastiCache attivati da IAM e convaliderà le richieste di connessione con IAM.

Limitazioni

Durante l'utilizzo dell'autenticazione IAM, valgono le seguenti limitazioni:

- L'autenticazione IAM è disponibile quando si utilizza ElastiCache per Redis versione 7.0 o successive.
- Per gli utenti ElastiCache attivati da IAM, le proprietà nome utente e ID utente devono essere identiche.
- 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 ElastiCache per Redis verrà disconnessa automaticamente 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 supporta le seguenti chiavi di contesto della condizione globale:
 - Quando si utilizza l'autenticazione IAM con cache serverless, `aws:VpcSourceIp`, `aws:SourceVpc`, `aws:SourceVpce`, `aws:CurrentTime`, `aws:EpochTime` e `aws:ResourceTag/%s` (dalle cache serverless e dagli utenti associati) sono supportate.
 - Quando si utilizza l'autenticazione IAM con i gruppi di replica, `aws:SourceIp` e `aws:ResourceTag/%s` (dai gruppi di replica e dagli utenti associati) sono supportate.

Per ulteriori informazioni sulle chiavi di contesto delle condizioni globali, consultare [Chiavi di contesto delle condizioni globali AWS](#) nella Guida per l'utente di IAM.

Installazione

Per impostare l'autenticazione IAM:

1. Crea una cache.

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name cache-01 \  
  --description "ElastiCache IAM auth application" \  
  --engine redis
```

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",  
      "Action" : [  
        "elasticache:Connect"  
      ],  
      "Resource" : [  
        "arn:aws:elasticache:us-east-1:123456789012:serverlesscache:cache-01",  
        "arn:aws:elasticache:us-east-1:123456789012:user:iam-user-01"  
      ]  
    }  
  ]  
}
```

4. Crea un ruolo IAM.

```
aws iam create-role \  
--role-name "elasticache-iam-auth-app" \  
--assume-role-policy-document file://trust-policy.json
```

5. Creare la policy IAM.

```
aws iam create-policy \  
--policy-name "elasticache-allow-all" \  
--policy-document file://policy.json
```

6. Allegare la policy IAM al ruolo.

```
aws iam attach-role-policy \  
--role-name "elasticache-iam-auth-app" \  
--policy-arn "arn:aws:iam::123456789012:policy/elasticache-allow-all"
```

7. Crea un nuovo utente attivato da IAM.

```
aws elasticache create-user \  
--user-name iam-user-01 \  
--user-id iam-user-01 \  
--authentication-mode Type=iam \  
--engine redis \  
--access-string "on ~* +@all"
```

8. Crea un gruppo di utenti e collega l'utente.

```
aws elasticache create-user-group \  
--user-group-id iam-user-group-01 \  
--engine redis \  
--user-ids default iam-user-01  
  
aws elasticache modify-serverless-cache \  
--serverless-cache-name cache-01 \  
--user-group-id iam-user-group-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 una cache Redis, come mostrato nell'esempio riportato di seguito.

```
String userId = "insert user id";
String cacheName = "insert cache name";
boolean isServerless = true;
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 ElastiCache
// Redis.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
    region, isServerless);
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(userId, iamAuthToken)
    .build();

// Create a new Lettuce Redis client
RedisClient client = RedisClient.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 PARAM_RESOURCE_TYPE = "ResourceType";
private static final String RESOURCE_TYPE_SERVERLESS_CACHE = "ServerlessCache";
private static final String ACTION_NAME = "connect";
private static final String SERVICE_NAME = "elasticache";
private static final long TOKEN_EXPIRY_SECONDS = 900;

private final String userId;
private final String cacheName;
private final String region;
private final boolean isServerless;

public IAMAuthTokenRequest(String userId, String cacheName, String region, boolean
isServerless) {
    this.userId = userId;
    this.cacheName = cacheName;
    this.region = region;
    this.isServerless = isServerless;
}

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(userId));
    if (isServerless) {
        request.addParameters(PARAM_RESOURCE_TYPE,
Collections.singletonList(RESOURCE_TYPE_SERVERLESS_CACHE));
    }
    return request;
}
```

```

private URI getRequestUri() {
    return URI.create(String.format("%s%s/", REQUEST_PROTOCOL, cacheName));
}

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 ElastiCache per Redis utilizzando il provider di credenziali dell'autenticazione IAM.

```

String userId = "insert user id";
String cacheName = "insert cache name";
boolean isServerless = true;
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 ElastiCache Redis.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
    region, isServerless);

```

```
// Create a Redis credentials provider using IAM credentials.
RedisCredentialsProvider redisCredentialsProvider = new
    RedisIAMAAuthCredentialsProvider(
        userId, 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 client
RedisClient client = RedisClient.create(redisURI);
client.connect();
```

Di seguito è riportato un esempio di client Lettuce Redis che racchiude `IAMAAuthTokenRequest` in un fornitore di credenziali per generare automaticamente credenziali temporanee quando richiesto.

```
public class RedisIAMAAuthCredentialsProvider implements RedisCredentialsProvider {
    private static final long TOKEN_EXPIRY_SECONDS = 900;

    private final AWSCredentialsProvider awsCredentialsProvider;
    private final String userId;
    private final IAMAAuthTokenRequest iamAuthTokenRequest;
    private final Supplier<String> iamAuthTokenSupplier;

    public RedisIAMAAuthCredentialsProvider(String userId,
        IAMAAuthTokenRequest iamAuthTokenRequest,
        AWSCredentialsProvider awsCredentialsProvider) {
        this.userName = userId;
        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(userId, iamAuthTokenSupplier.get()));
    }
}
```

```
    }

    private String getIamAuthToken() {
        return
iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());
    }
}
```

Autenticazione con il comando Redis AUTH

Note

Il comando AUTH Redis è stato sostituito da [the section called “Controllo accessi basato sui ruoli \(RBAC\)”](#). Tutte le cache serverless devono utilizzare il controllo degli accessi basato su ruolo (RBAC) per l'autenticazione.

Questi token o password di autenticazione consentono a Redis di richiedere una password prima di permettere ai clienti di eseguire i comandi e migliorare così la sicurezza dei dati. Il comando AUTH Redis è disponibile solo per i cluster progettati autonomamente.

Argomenti

- [Panoramica di AUTH in ElastiCache for Redis](#)
- [Applicazione dell'autenticazione a un cluster ElastiCache for Redis](#)
- [Modifica del token AUTH su un cluster ElastiCache for Redis esistente](#)
- [Migrazione da RBAC a Redis AUTH](#)

Panoramica di AUTH in ElastiCache for Redis

Quando utilizzi Redis AUTH con il cluster ElastiCache for Redis, hai a disposizione diverse migliorie.

In particolare, considera le seguenti limitazioni dei token , o password AUTH quando utilizzi AUTH con ElastiCache pe Redis:

- I token o password devono essere composti da 16–128 caratteri stampabili.
- I caratteri non alfanumerici sono limitati a (!, &, #, \$, ^, <, >, -).
- AUTH può essere abilitato solo per i cluster ElastiCache for Redis abilitati alla crittografia in transito.

Per configurare un token robusto, si consiglia di seguire una policy per password rigida che richieda, ad esempio, quanto segue:

- I token o le password devono includere almeno tre dei seguenti tipi di carattere:
 - Caratteri maiuscoli
 - Caratteri minuscoli
 - Numeri
 - Caratteri non alfanumerici (!, &, #, \$, ^, <, >, -)
- I token o password non devono contenere una parola di senso compiuto o una parola di senso compiuto leggermente modificata.
- I token o password non devono essere uguali o simili a un token utilizzato di recente.

Applicazione dell'autenticazione a un cluster ElastiCache for Redis

È possibile richiedere che gli utenti inseriscano un token (password) in un server Redis protetto da token. Per fare ciò, includi il parametro `--auth-token` (API: `AuthToken`) con il token corretto quando crei il gruppo di replica o il cluster. Includerlo anche in tutti i comandi successivi per il gruppo di replica o il cluster.

La seguente operazione AWS CLI crea un gruppo di replica con crittografia in transito (TLS) abilitata e il token AUTH *This-is-a-sample-token*. Sostituisci il gruppo di sottoreti `sng-test` con uno esistente.

Parametri chiave

- **--engine** — Deve essere `redis`.
- **--engine-version** - Deve essere 3.2.6, 4.0.10 o versione successiva.
- **--transit-encryption-enabled** : obbligatorio per l'autenticazione e la conformità HIPAA.
- **--auth-token** : obbligatorio per la conformità HIPAA. Questo valore deve essere il token corretto per questo server Redis protetto da token.
- **--cache-subnet-group** : obbligatorio per la conformità HIPAA.

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group \
```



```
--replication-group-id authtestgroup \  
--replication-group-description authtest \  
--engine redis \  
--cache-node-type cache.m4.large \  
--num-node-groups 1 \  
--replicas-per-node-group 2 \  
--transit-encryption-enabled \  
--auth-token This-is-a-sample-token \  
--cache-subnet-group sng-test
```

Per Windows:

```
aws elasticache create-replication-group ^  
--replication-group-id authtestgroup ^  
--replication-group-description authtest ^  
--engine redis ^  
--cache-node-type cache.m4.large ^  
--num-node-groups 1 ^  
--replicas-per-node-group 2 ^  
--transit-encryption-enabled ^  
--auth-token This-is-a-sample-token ^  
--cache-subnet-group sng-test
```

Modifica del token AUTH su un cluster ElastiCache for Redis esistente

Per semplificare l'aggiornamento dell'autenticazione, è possibile modificare il token AUTH utilizzato su un cluster ElastiCache for Redis. È possibile apportare questa modifica se la versione del motore è la 5.0.6 o superiore e se in ElastiCache per Redis la crittografia dei dati in transito è abilitata.

La modifica del token di autorizzazione supporta due strategie: ROTATE e SET. La strategia ROTATE aggiunge un token AUTH ulteriore al server conservando al contempo il token precedente. La strategia SET aggiorna il server per supportare solo un singolo token AUTH. Effettuare queste richieste di modifica con il parametro `--apply-immediately` per applicare immediatamente le modifiche.

Rotazione del token AUTH

Per aggiornare un server Redis con un nuovo token AUTH, richiamare l'API

`ModifyReplicationGroup` con il parametro `--auth-token` come nuovo token di autorizzazione e `--auth-token-update-strategy` con il valore ROTATE. Al completamento della modifica, il cluster supporterà sia il token AUTH precedente sia quello specificato nel parametro `auth-token`.

Note

Se prima non si configura il token AUTH, una volta completata la modifica, il cluster non supporterà alcun token AUTH oltre a quello specificato nel parametro `auth-token`.

Se questa modifica viene eseguita su un server che già supporta due token AUTH, il token AUTH meno recente viene rimosso durante questa operazione per permettere al server di supportare sempre un massimo di due token AUTH più recenti.

A questo punto, è possibile procedere aggiornando il client per utilizzare l'ultimo token AUTH. Una volta aggiornati i client, è possibile utilizzare la strategia SET in modo che la rotazione del token AUTH (spiegata nella sezione seguente) inizi esclusivamente utilizzando il nuovo token.

L'operazione AWS CLI seguente modifica il gruppo di replica per ruotare il token AUTH *This-is-the-rotated-token*.

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \  
--replication-group-id authtestgroup \  
--auth-token This-is-the-rotated-token \  
--auth-token-update-strategy ROTATE \  
--apply-immediately
```

Per Windows:

```
aws elasticache modify-replication-group ^  
--replication-group-id authtestgroup ^  
--auth-token This-is-the-rotated-token ^  
--auth-token-update-strategy ROTATE ^  
--apply-immediately
```

Impostazione del token AUTH

Per aggiornare un server Redis con due token AUTH in modo che supporti un solo token AUTH, richiamare l'operazione API `ModifyReplicationGroup`. Richiamare `ModifyReplicationGroup` con il parametro `--auth-token` come nuovo token AUTH e il parametro `--auth-token-update-strategy` con il valore SET. Il parametro `auth-token` deve essere lo stesso valore del token

AUTH ruotato. Una volta completata la modifica, il server Redis supporta solo il token AUTH specificato nel parametro `auth-token`.

L'operazione AWS CLI seguente modifica il gruppo di replica per impostare il token AUTH su *This-is-the-set-token*.

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \  
--replication-group-id authtestgroup \  
--auth-token This-is-the-set-token \  
--auth-token-update-strategy SET \  
--apply-immediately
```

Per Windows:

```
aws elasticache modify-replication-group ^  
--replication-group-id authtestgroup ^  
--auth-token This-is-the-set-token ^  
--auth-token-update-strategy SET ^  
--apply-immediately
```

Abilitazione dell'autenticazione su un cluster ElastiCache for Redis esistente

Per abilitare l'autenticazione su un server Redis esistente, richiamare l'operazione API `ModifyReplicationGroup`. Richiamare `ModifyReplicationGroup` con il parametro `--auth-token` come il nuovo token e `--auth-token-update-strategy` con il valore `ROTATE`.

Una volta completata la modifica, il cluster supporta il token AUTH specificato nel parametro `auth-token`, oltre a supportare la connessione senza autenticazione. L'abilitazione dell'autenticazione è supportata solo sui server Redis con crittografia in transito abilitata (TLS).

Migrazione da RBAC a Redis AUTH

Se si autenticano gli utenti con Redis Role-Based Access Control (RBAC) come descritto in [Autenticazione degli utenti con controllo degli accessi basato sui ruoli \(RBAC\)](#) e desidera eseguire la migrazione a Redis AUTH, utilizzare le seguenti procedure. È possibile eseguire la migrazione mediante console o CLI.

Per eseguire la migrazione da RBAC a Redis AUTH utilizzando la console

1. Accedi alla AWS Management Console e apri la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dall'elenco nell'angolo in alto a destra, scegli la regione AWS dove si trova il cluster da modificare.
3. Nel riquadro di navigazione, scegliere il motore in esecuzione sul cluster da modificare.

Comparirà un elenco dei cluster che eseguono il motore selezionato.

4. Nell'elenco dei cluster, per il cluster che si desidera modificare, scegli il nome.
5. Per Operazioni, scegli Modifica.

Viene visualizzata la finestra Modifica.

6. Per Controllo accessi scegli Accesso utente predefinito AUTH Redis.
7. Per Token AUTH Redis imposta un nuovo token.
8. Scegli Visualizza l'anteprima delle modifiche e quindi, nella schermata successiva, seleziona Modifica.

Per eseguire la migrazione da RBAC a Redis AUTH utilizzando AWS CLI

- Utilizzare uno dei seguenti comandi.

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id test \  
  --remove-user-groups \  
  --auth-token password \  
  --auth-token-update-strategy SET \  
  --apply-immediately
```

Per Windows:

```
aws elasticache modify-replication-group ^\  
  --replication-group-id test ^\  
  --remove-user-groups ^\  
  --auth-token password ^\  
  --auth-token-update-strategy SET ^
```

```
--apply-immediately
```

Per ulteriori informazioni sull'utilizzo di AUTH, consulta [Token AUTH](#) sul sito Web [redis.io](#).

Note

Se è necessario disabilitare il controllo degli accessi su un cluster ElastiCache, consulta [the section called “Disabilitazione del controllo degli accessi su una cache ElastiCache Redis”](#)

Disabilitazione del controllo degli accessi su una cache ElastiCache Redis

Segui le istruzioni riportate di seguito per disabilitare il controllo degli accessi su una cache Redis abilitata per TLS. La cache Redis è stata configurata in uno dei due diversi tipi disponibili: accesso utente predefinito AUTH Redis o lista di controllo degli accessi (RBAC) dei gruppi di utenti. Se la cache è stata creata con la configurazione AUTH, è necessario modificarla nella configurazione RBAC prima di poterla disabilitare rimuovendo i gruppi di utenti. Se la cache è stata creata con la configurazione RBAC, puoi procedere direttamente alla disabilitazione.

Per disabilitare una cache serverless Redis configurata con RBAC

1. Rimuovi i gruppi di utenti per disabilitare il controllo degli accessi.

```
aws elasticache modify-serverless-cache --serverless-cache-name <serverless-cache>
--remove-user-group
```

2. (Facoltativo) Verifica che nessun gruppo di utenti sia associato alla cache serverless.

```
aws elasticache describe-serverless-caches --serverless-cache-name <serverless-
cache>
{
  "...",
  "UserGroupId": "",
  "...",
}
```

Per disabilitare una cache Redis configurata con un token AUTH

1. Cambia il token AUTH in RBAC e specifica un gruppo di utenti da aggiungere.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-id-value> --auth-token-update-strategy DELETE --user-group-ids-to-add <user-group-value>
```

2. Verifica che il token AUTH sia stato disabilitato e che sia stato aggiunto un gruppo di utenti.

```
aws elasticache describe-replication-groups --replication-group-id <replication-group-id-value>
{
  "...
  "AuthTokenEnabled": false,
  "UserGroupIds": [
    "<user-group-value>"
  ]
  "...
}
```

3. Rimuovi i gruppi di utenti per disabilitare il controllo degli accessi.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-value> --user-group-ids-to-remove <user-group-value>
{
  "...
  "PendingModifiedValues": {
    "UserGroups": {
      "UserGroupIdsToAdd": [],
      "UserGroupIdsToRemove": [
        "<user-group-value>"
      ]
    }
  }
  "...
}
```

4. (Facoltativo) Verifica che nessun gruppo di utenti sia associato al cluster. Il campo `AuthTokenEnabled` deve essere impostato su `false`.

```
aws elasticache describe-replication-groups --replication-group-id <replication-group-value>
"AuthTokenEnabled": false
```

Per disabilitare un cluster Redis configurato con RBAC

1. Rimuovi i gruppi di utenti per disabilitare il controllo degli accessi.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-value> --user-group-ids-to-remove <user-group-value>
{
  "...
  "PendingModifiedValues": {
    "UserGroups": {
      "UserGroupIdsToAdd": [],
      "UserGroupIdsToRemove": [
        "<user-group-value>"
      ]
    }
  }
  "...
}
```

2. (Facoltativo) Verifica che nessun gruppo di utenti sia associato al cluster. Il campo `AuthTokenEnabled` deve essere impostato su `false`.

```
aws elasticache describe-replication-groups --replication-group-id <replication-group-value>
"AuthTokenEnabled": false
```

Riservatezza del traffico Internet

Amazon ElastiCache utilizza le seguenti tecniche per rendere sicuri i dati nella cache e proteggerli dagli accessi non autorizzati:

- [Amazon VPC e sicurezza di ElastiCache](#) spiega il tipo di gruppo di sicurezza necessario per l'installazione.
- [Identity and Access Management per Amazon ElastiCache](#) per concedere e limitare le operazioni di utenti, gruppi e ruoli.

Amazon VPC e sicurezza di ElastiCache

Poiché la sicurezza dei dati è importante, ElastiCache offre i mezzi per controllare chi ha accesso ai tuoi dati. Il modo in cui controlli l'accesso ai tuoi dati dipende dal fatto che tu abbia avviato o meno i cluster in un Virtual Private Cloud (Amazon VPC) di Amazon o Amazon EC2-Classic.

Important

Abbiamo reso obsoleto l'uso di Amazon EC2-Classic per avviare i cluster di ElastiCache. Tutti i nodi della generazione corrente vengono avviati solo in Amazon Virtual Private Cloud.

Il servizio Virtual Private Cloud (Amazon VPC) di Amazon definisce una rete virtuale che ricorda molto un data center tradizionale. Quando vuoi configurare Amazon VPC, puoi selezionare l'intervallo di indirizzi IP, creare sottoreti e configurare tabelle di routing, gateway di rete e le impostazioni di sicurezza. Puoi anche aggiungere un cluster di cache alla rete virtuale e controllare l'accesso al cluster di cache usando i gruppi di sicurezza di Amazon VPC.

Questa sezione spiega come configurare manualmente un cluster di ElastiCache in un Amazon VPC. Queste informazioni sono rivolte agli utenti che desiderano comprendere come interagiscono ElastiCache e Amazon VPC.

Argomenti

- [Informazioni su ElastiCache e Amazon VPC](#)
- [Modelli di accesso a una cache ElastiCache in un Amazon VPC](#)
- [Creazione di un virtual private cloud \(VPC\).](#)
- [Connessione a una cache in esecuzione in un Amazon VPC](#)

Informazioni su ElastiCache e Amazon VPC

ElastiCache è completamente integrato con il Virtual Private Cloud (Amazon VPC) di Amazon. Per gli utenti di ElastiCache, questo significa che:

- Se l'account AWS supporta solo la piattaforma EC2-VPC, ElastiCache avvia sempre il cluster in un Amazon VPC.
- Se non hai mai utilizzato AWS, i tuoi cluster saranno implementati in un Amazon VPC. Verrà creato automaticamente un VPC di default.
- 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](#).

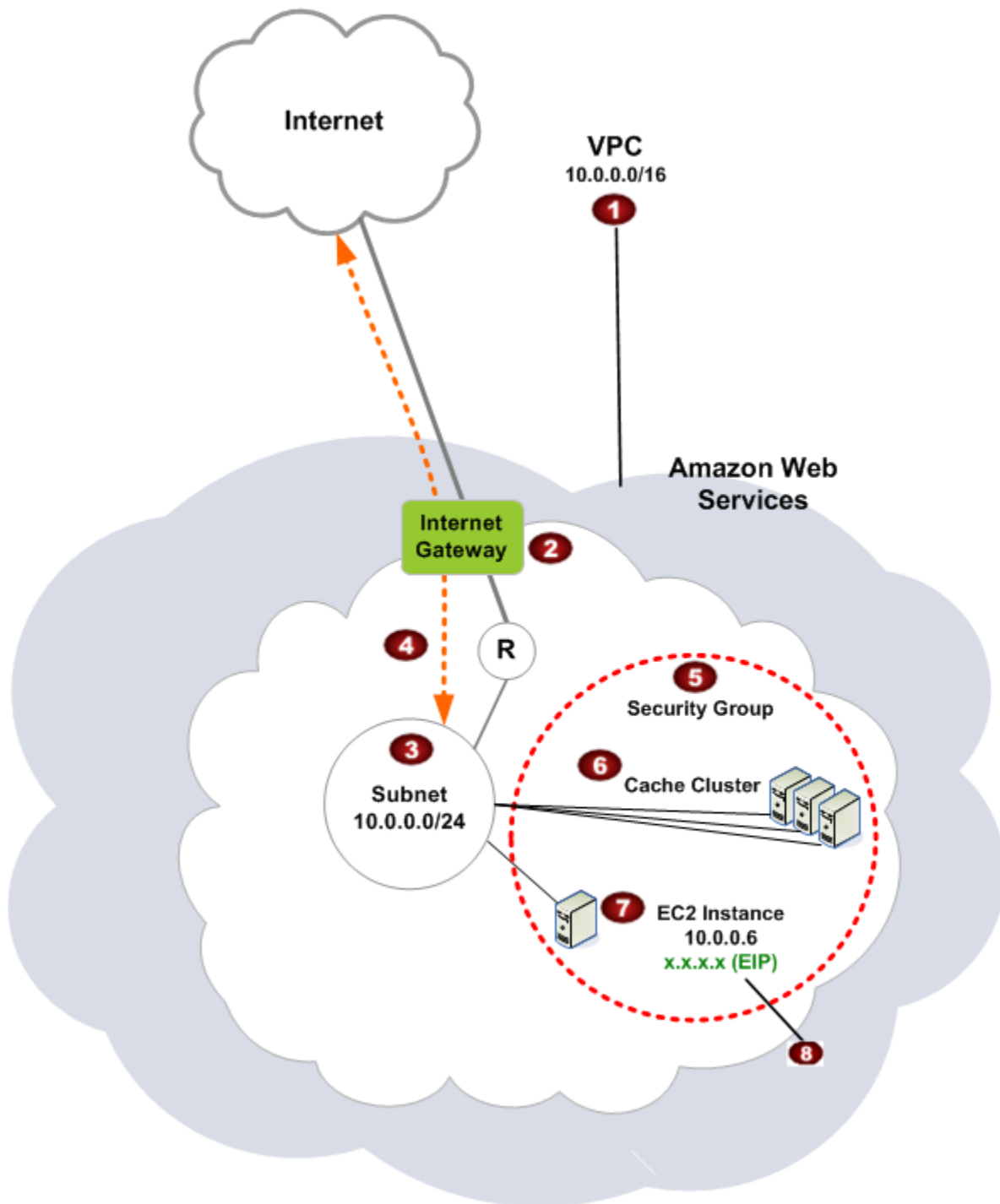
Con Amazon Virtual Private Cloud puoi creare una rete virtuale nel AWS cloud che ricorda molto un data center tradizionale. Puoi configurare il Amazon VPC, inclusi la selezione degli intervalli di indirizzi IP, la creazione di sottoreti, la configurazione di tabelle di routing, i gateway di rete, nonché le impostazioni di sicurezza.

La funzionalità di base di ElastiCache è la stessa di quella di un cloud privato virtuale; ElastiCache gestisce gli aggiornamenti del software, le patch, il rilevamento dell'errore e il ripristino a seconda che i cluster siano implementati all'interno o all'esterno di una Amazon VPC.

I nodi di cache di ElastiCache implementati all'esterno di una Amazon VPC vengono assegnati a un indirizzo IP sul quale si risolve il nome DNS/endpoint. Ciò consente la connettività dalle istanze di Amazon Elastic Compute Cloud (Amazon EC2). Quando avvii un cluster di ElastiCache in una sottorete privata di Amazon VPC, a ogni nodo di cache viene assegnato un indirizzo IP privato all'interno di tale sottorete.

Panoramica di ElastiCache in una Amazon VPC

Il diagramma e la tabella seguenti descrivono l'ambiente di Amazon VPC insieme ai cluster di ElastiCache e alle istanze Amazon EC2 che vengono avviate nella Amazon VPC.

**1**

L'Amazon VPC è una porzione isolata della AWS Cloud a cui è stato assegnato il proprio blocco di indirizzi IP.

2

Un gateway Internet collega la Amazon VPC direttamente a Internet e fornisce accesso alle altre risorse AWS quali Amazon Simple Storage Service (Amazon S3) in esecuzione al di fuori della Amazon VPC.

3

Una sottorete di Amazon VPC è un segmento dell'intervallo di indirizzi IP di una Amazon VPC in cui è possibile isolare le risorse AWS in base alle esigenze operative e di sicurezza.

4

Una tabella di routing nella Amazon VPC instrada il traffico di rete tra la sottorete e Internet. L'Amazon VPC ha un router implicito che, in questo diagramma, è rappresentato dal simbolo del cerchio con la R.

5

Un gruppo di sicurezza di Amazon VPC controlla il traffico in entrata e in uscita per i cluster di ElastiCache e le istanze Amazon EC2.

6

È possibile avviare un cluster di ElastiCache nella sottorete. I nodi della cache 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 di cache nella stessa sottorete.

8

Affinché un'istanza Amazon EC2 nella Amazon VPC sia raggiungibile da Internet, è necessario assegnare un indirizzo pubblico, statico, chiamato indirizzo IP elastico all'istanza.

Prerequisiti

Per creare un cluster di ElastiCache all'interno di una Amazon VPC, l'Amazon VPC deve soddisfare i seguenti requisiti:

- L'Amazon VPC deve consentire istanze Amazon EC2 non dedicate. Non puoi utilizzare ElastiCache in una Amazon VPC configurata per una locazione dell'istanza dedicata.

- È necessario definire un gruppo di sottorete di cache per la Amazon VPC. ElastiCache utilizza quel gruppo di sottorete di cache per selezionare una sottorete e gli indirizzi IP all'interno di quella sottorete da associare agli endpoint VPC o ai nodi di cache.
- I blocchi CIDR per ogni sottorete devono essere abbastanza grandi da fornire indirizzi IP di riserva affinché ElastiCache possa utilizzarli durante le attività di manutenzione.

Routing e sicurezza

È possibile configurare il routing nell'Amazon VPC per controllare i punti di concentrazione dei flussi di traffico (ad esempio, il gateway Internet o il gateway virtuale privato). Con un gateway Internet, l'Amazon VPC accede direttamente alle altre risorse AWS che non sono in esecuzione nell'Amazon VPC. Se scegliere 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 alle risorse AWS in Internet.

Puoi utilizzare i gruppi di sicurezza di Amazon VPC per contribuire a proteggere i cluster di ElastiCache 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 della cache perché l'indirizzo IP sottostante può cambiare.

Documentazione Amazon VPC

Amazon VPC dispone della propria documentazione che descrive come creare e usare l'Amazon VPC. La tabella seguente fornisce i collegamenti alle 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

Descrizione	Documentazione
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 una cache ElastiCache in un Amazon VPC

Amazon ElastiCache supporta i seguenti scenari per accedere a una cache in un Amazon VPC:

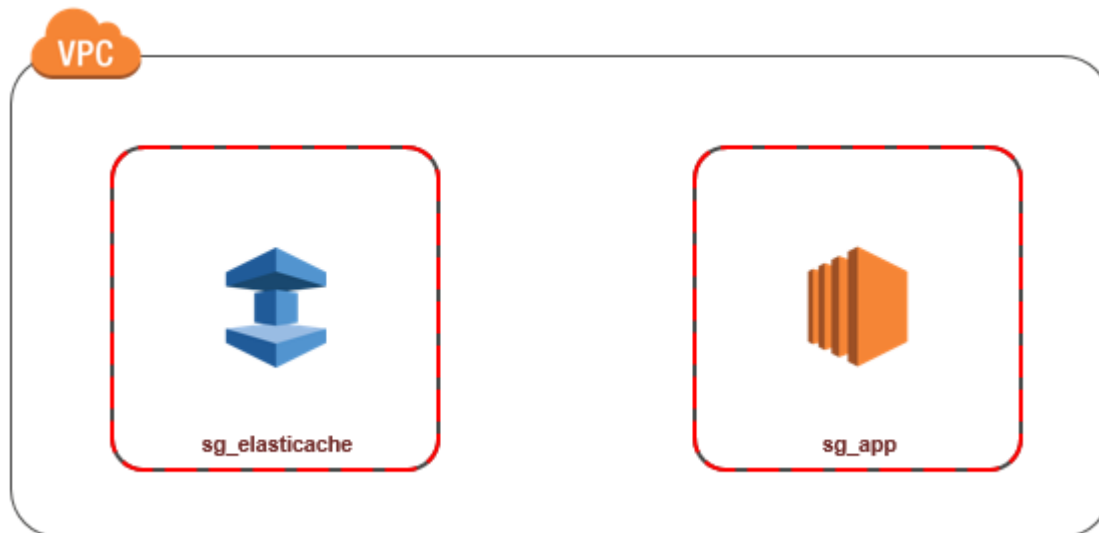
Indice

- [Accesso a una cache ElastiCache quando la cache e l'istanza Amazon EC2 sono nello stesso Amazon VPC](#)
- [Accesso a una cache ElastiCache quando la cache e l'istanza Amazon EC2 sono in differenti Amazon VPC](#)
 - [Accesso a una cache ElastiCache quando la cache e l'istanza Amazon EC2 sono in differenti Amazon VPC nella stessa regione](#)
 - [Uso del Transit Gateway](#)
 - [Accesso a una cache ElastiCache quando la cache e l'istanza Amazon EC2 sono in Amazon VPC differenti in regioni differenti](#)
 - [Utilizzo di VPC di transito](#)
- [Accesso a una cache ElastiCache da un'applicazione in esecuzione nel data center di un cliente](#)
 - [Accesso a una cache ElastiCache da un'applicazione in esecuzione nel data center di un cliente tramite connessione VPN](#)
 - [Accesso a una cache ElastiCache da un'applicazione in esecuzione nel data center di un cliente tramite Direct Connect](#)

Accesso a una cache ElastiCache quando la cache e l'istanza Amazon EC2 sono nello stesso Amazon VPC

Il caso d'uso più comune è quando un'applicazione implementata su un'istanza EC2 deve connettersi a una cache nello stesso VPC.

Nel seguente diagramma viene illustrato questo scenario:



Il modo più semplice per gestire l'accesso tra istanze EC2 e cache nello stesso VPC è procedere come segue:

1. Crea un gruppo di sicurezza VPC per la cache. Questo gruppo di sicurezza può essere usato per limitare l'accesso alla cache. Per questo gruppo di sicurezza è, ad esempio, possibile creare una regola personalizzata che consenta l'accesso TCP tramite la porta assegnata alla cache al momento della creazione e un indirizzo IP che viene utilizzato per accedere alla cache.

La porta predefinita per le cache Redis 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 le regole personalizzate nel gruppo di sicurezza per la cache che consentano le connessioni dal gruppo di sicurezza creato per le istanze EC2. In tal modo qualsiasi membro del gruppo di sicurezza può accedere alle cache.

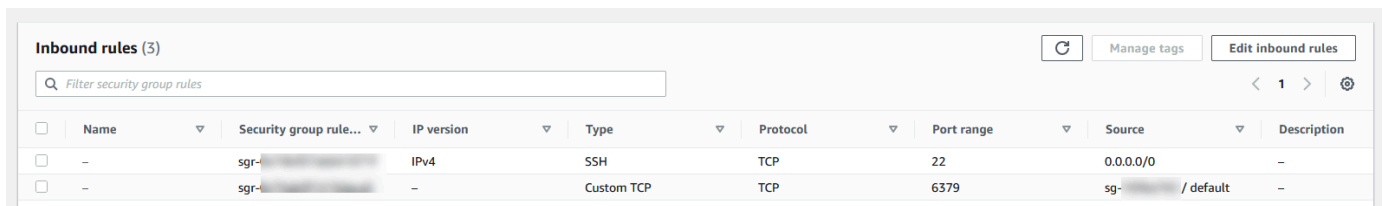
Note

Se si prevede di utilizzare [Local Zones](#), assicurati di averlo abilitato. Quando si crea un gruppo di sottorete in quella zona locale, la VPC viene estesa a quella zona locale e la VPC considera la sottorete come qualsiasi sottorete in qualsiasi altra zona di disponibilità. Tutti i gateway e le tabelle di routing pertinenti verranno regolati automaticamente.

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. Fare clic su Security Groups (Gruppi di sicurezza) nel riquadro di navigazione.
3. Scegli o crea un gruppo di sicurezza da utilizzare per la cache. 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 Intervallo di porte specifica la porta utilizzata alla creazione della cache.

La porta predefinita per le cache Redis e i gruppi di replica 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.



<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sg-...	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sg-...	-	Custom TCP	TCP	6379	sg-... / default	-

Accesso a una cache ElastiCache quando la cache e l'istanza Amazon EC2 sono in differenti Amazon VPC

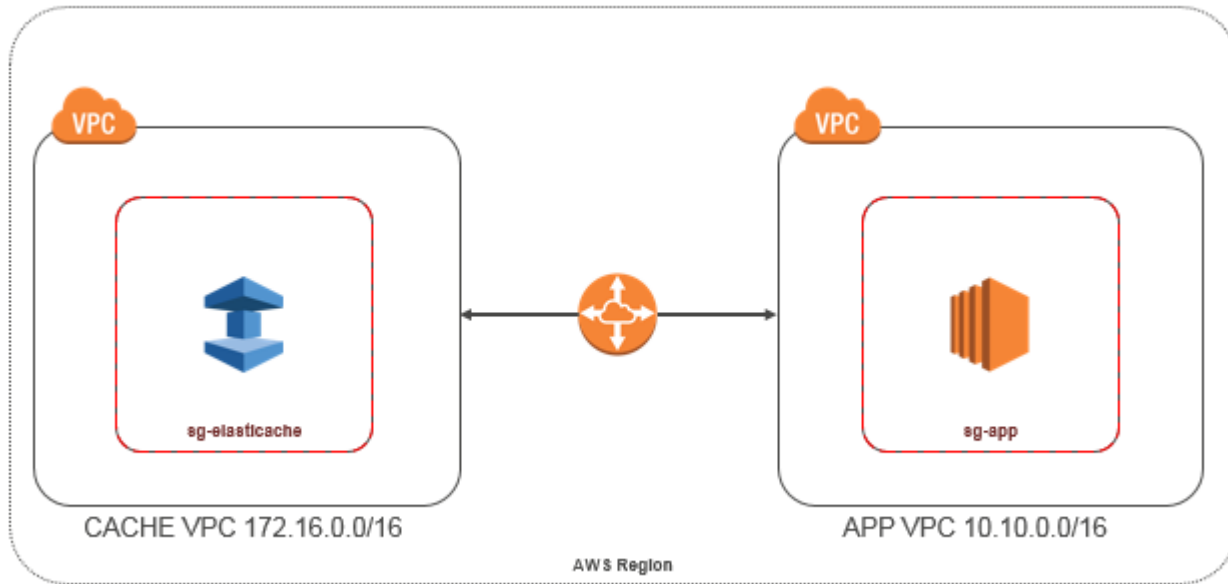
Quando la cache si trova in un VPC differente da quello in cui si trova l'istanza EC2 che utilizzi per accedervi, è possibile accedere alla cache in diversi modi. Se la cache e l'istanza EC2 si trovano in VPC differenti, ma nella stessa regione, puoi utilizzare il peering VPC. Se la cache e l'istanza EC2 si trovano in regioni differenti, puoi creare una connessione VPN tra le regioni.

Argomenti

- [Accesso a una cache ElastiCache quando la cache e l'istanza Amazon EC2 sono in differenti Amazon VPC nella stessa regione](#)
- [Accesso a una cache ElastiCache quando la cache e l'istanza Amazon EC2 sono in Amazon VPC differenti in regioni differenti](#)

Accesso a una cache ElastiCache quando la cache e l'istanza Amazon EC2 sono in differenti Amazon VPC nella stessa regione

Il seguente diagramma illustra l'accesso a una cache da un'istanza Amazon EC2 in un Amazon VPC differente presente nella stessa regione, usando una connessione peering Amazon VPC.



Cache a cui si accede da un'istanza Amazon EC2 in un 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](#).

Note

La risoluzione dei nomi DNS potrebbe non riuscire per i VPC con peering, a seconda delle configurazioni applicate al VPC ElastiCache. Per risolvere tale problema, entrambi i VPC devono essere abilitati per nomi host DNS e risoluzione DNS. Per ulteriori informazioni, consulta [Abilitazione della risoluzione DNS per una connessione peering VPC](#).

Per accedere a una cache in un Amazon VPC differente con 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](#).

Di seguito è mostrato l'aspetto delle tabelle di routing relative all'esempio nel diagramma precedente. `pcx-a894f1c1` è la connessione peering.

Destination	Target	Destination	Target
172.16.0.0/16	local	10.10.0.0/16	local
10.10.0.0/16	pcx-a894f1c1	0.0.0.0/0	igw-bfdcccd8
		172.16.0.0/16	pcx-a894f1c1

Tabella di routing VPC

4. Modifica il gruppo di sicurezza della cache ElastiCache per consentire la connessione in entrata del 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 una cache con una connessione peering implica 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.

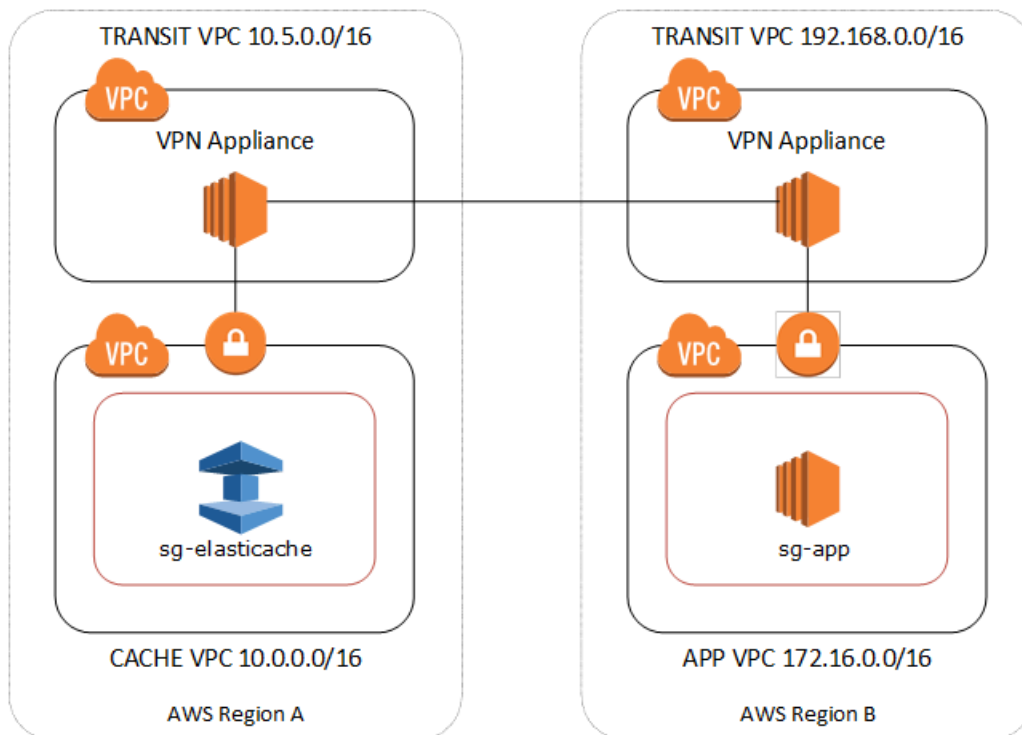
È inoltre possibile creare un allegato di connessione di peering tra gateway di transito in diverse regioni AWS. In questo modo è possibile instradare il traffico tra gli allegati dei gateway di transito in diverse regioni.

Per ulteriori informazioni, consulta [Gateway di transito](#).

Accesso a una cache ElastiCache quando la cache e l'istanza Amazon EC2 sono in Amazon VPC differenti 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 implementata in un VPC "spoke" in una regione può essere connessa a una cache ElastiCache in un VPC "spoke" all'interno di un'altra regione.

Per accedere a una cache 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 nei VPC di app e cache 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. Modifica il gruppo di sicurezza della cache ElastiCache 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 una cache tra regioni introduce latenze di rete e ulteriori costi di trasferimento dei dati tra regioni.

Accesso a una cache ElastiCache da un'applicazione in esecuzione nel data center di un cliente

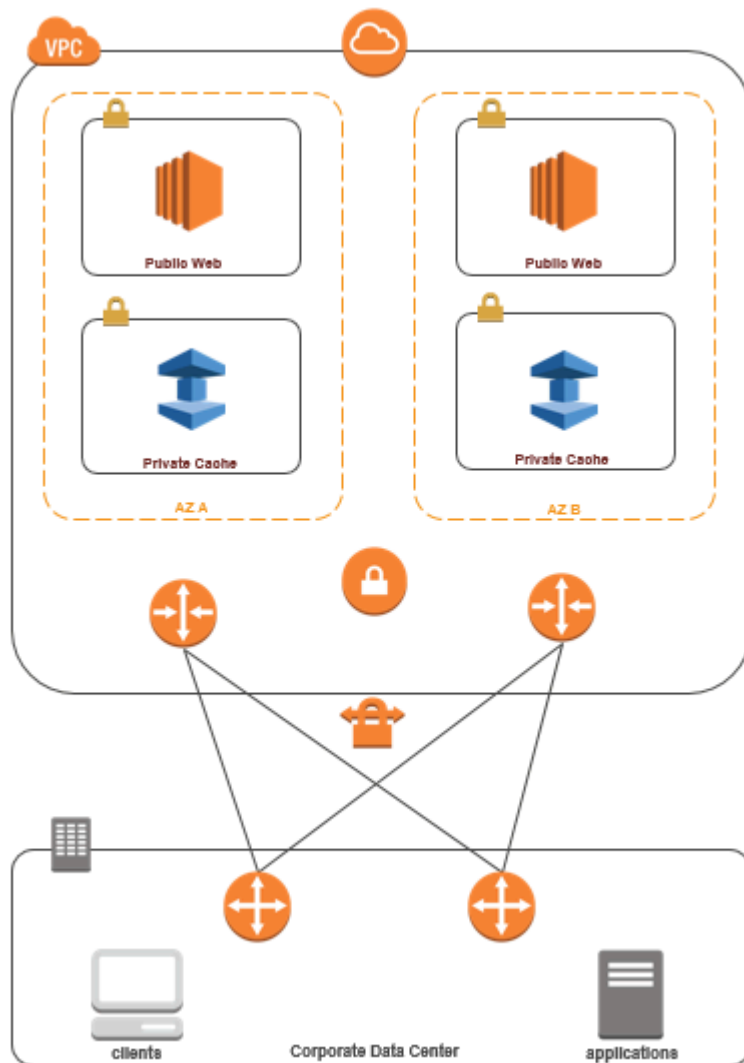
Un altro possibile scenario è rappresentato da un'architettura ibrida in cui i client o le applicazioni nel data center del cliente possono dover accedere a una cache ElastiCache nel VPC. Anche questo scenario è supportato purché sia disponibile una connessione tra VPC e data center dei clienti tramite VPN o Direct Connect.

Argomenti

- [Accesso a una cache ElastiCache da un'applicazione in esecuzione nel data center di un cliente tramite connessione VPN](#)
- [Accesso a una cache ElastiCache da un'applicazione in esecuzione nel data center di un cliente tramite Direct Connect](#)

Accesso a una cache ElastiCache da un'applicazione in esecuzione nel data center di un cliente tramite connessione VPN

Il diagramma seguente illustra l'accesso a una cache ElastiCache da un'applicazione in esecuzione nella rete aziendale tramite connessioni VPN.



Connessione a ElastiCache dal data center tramite una VPN

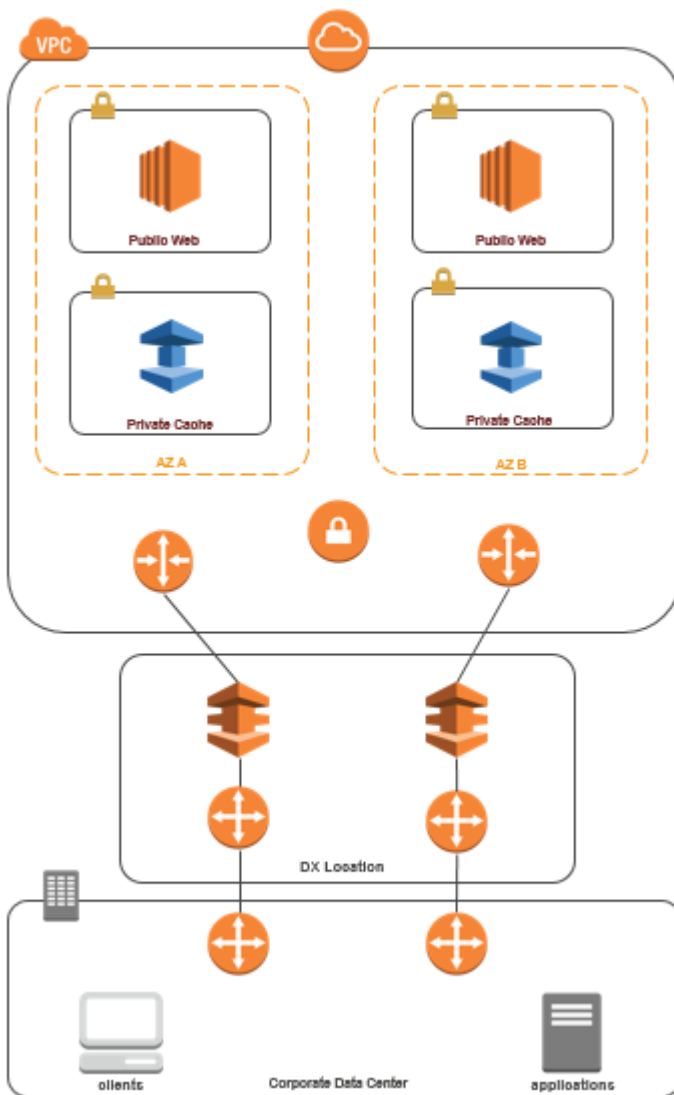
Per accedere a una cache in un VPC da un'applicazione on-premise con una connessione VPN

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. Aggiorna la tabella di routing del VPC per la sottorete in cui viene implementata la cache ElastiCache per consentire il traffico dal server di applicazioni on-premise. Nel caso di un routing dinamico con BGP le route possono essere automaticamente propagate.
3. Modifica il gruppo di sicurezza della cache ElastiCache per consentire la connessione in entrata dai server di applicazioni on-premise.

L'accesso a una cache con una connessione VPN introduce latenze di rete e ulteriori costi di trasferimento dei dati.

Accesso a una cache ElastiCache da un'applicazione in esecuzione nel data center di un cliente tramite Direct Connect

Il diagramma seguente illustra l'accesso a una cache ElastiCache da un'applicazione in esecuzione nella rete aziendale tramite Direct Connect.



Connessione a ElastiCache dal data center tramite Direct Connect

Per accedere a una cache ElastiCache da un'applicazione in esecuzione nella rete tramite Direct Connect

1. Stabilire una connessione Direct Connect. Per ulteriori informazioni, consulta [Nozioni di base su AWS Direct Connect](#).
2. Modifica il gruppo di sicurezza della cache ElastiCache per consentire la connessione in entrata dai server di applicazioni on-premise.

L'accesso a una cache con 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 Amazon VPC con una sottorete privata per ogni zona di disponibilità.

Creazione di un Amazon VPC (Console)

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 Crea 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, specificando quello pubblico e 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 di cache e l'istanza Amazon EC2.
 - a. Nel pannello di navigazione della console di gestione di Amazon VPC, scegliere Security Groups (Gruppi di sicurezza).
 - b. Scegli Crea gruppo di sicurezza.
 - c. Digita un nome e una descrizione per il gruppo di sicurezza nelle caselle corrispondenti. Nella casella VPC, seleziona l'identificativo del tuo Amazon VPC.

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name [Info](#)

Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

Inbound rules [Info](#)

This security group has no inbound rules.

[Add rule](#)

Outbound rules [Info](#)

Type	Protocol	Port range	Destination	Description - optional	
All traffic	All	All	Custom	<input type="text" value=""/>	Delete
			<input type="text" value="0.0.0.0"/>		

[Add rule](#)

- 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 pannello di navigazione, fare clic su 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).
 - d. Impostare i seguenti valori per la nuova regola in entrata per consentire l'accesso HTTP:
 - Tipo: HTTP
 - Fonte: 0.0.0.0/0

Scegliere Apply Rule Changes (Applica modifiche della regola).

Ora sei pronto per creare un gruppo di sottorete della cache e per avviare un cluster di cache nel tuo Amazon VPC.

- [Creazione di un gruppo di sottoreti](#)
- [Creazione di un cluster Redis \(modalità cluster disabilitata\) \(Console\)](#).

Connessione a una cache in esecuzione in un Amazon VPC

Questo esempio illustra come avviare un'istanza Amazon EC2 nel Amazon VPC. Puoi quindi accedere a questa istanza e alla cache ElastiCache in esecuzione in Amazon VPC.

Connessione a una cache in esecuzione in un Amazon VPC (console)

In questo esempio, si crea un'istanza Amazon EC2 nell'Amazon VPC. Puoi utilizzare questa istanza Amazon EC2 per connetterti ai nodi di cache in esecuzione nell'Amazon VPC.

Note

Per informazioni sull'uso di Amazon EC2, consulta la [Guida alle operazioni di base di Amazon EC2](#) nella [Documentazione relativa ad Amazon EC2](#).

Per creare un'istanza Amazon EC2 nel Amazon VPC tramite la console di Amazon EC2

1. Accedi a AWS Management Console e apri la console Amazon EC2 all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Nella console, scegliere Avvia istanza e attenersi alla seguente procedura:
3. Nella pagina scegliere Amazon Machine Image (AMI), scegliere un'AMI Amazon Linux a 64 bit, quindi scegliere Seleziona.
4. Nella pagina Scegliere un tipo di istanza, scegliere 3. Configura l'istanza.
5. Nella pagina Configura dettagli istanza effettuare le seguenti selezioni:
 - a. Nell'elenco Rete scegli il tuo Amazon VPC.
 - b. Nell'elenco Sottoreti scegliere la sottorete pubblica.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage pricing, assign an access management role to the instance, and more.

Number of instances ⓘ 1

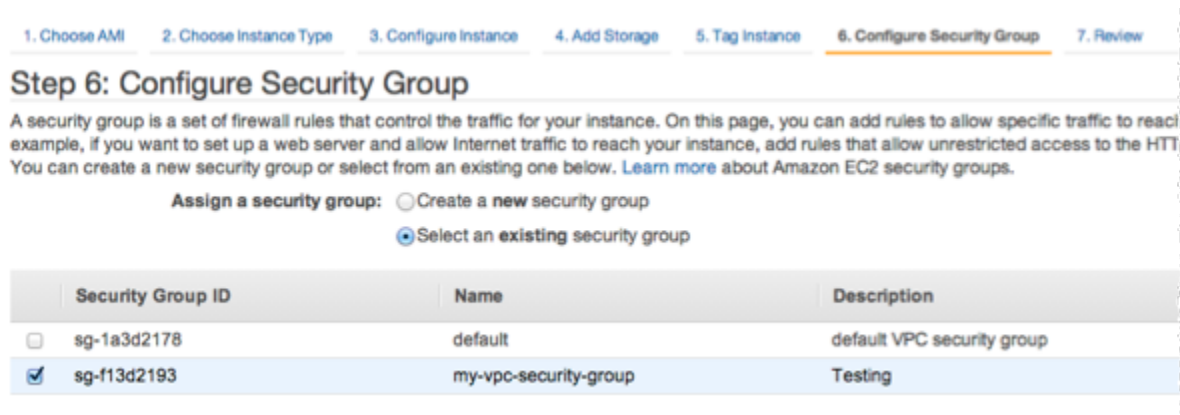
Purchasing option ⓘ Request Spot Instances

Network ⓘ vpc-d3a77cb6 (10.0.0.0/16) :

Subnet ⓘ subnet-58f5e63a(10.0.0.0/24) | sa-east-1a :
250 IP Addresses available

Public IP ⓘ Automatically assign a public IP address to your instances

- Dopo aver selezionato le impostazioni desiderate, scegliere 4. Aggiungi storage.
- Nella pagina Aggiungi archiviazione scegliere 5. Assegna un tag all'istanza.
 - Nella pagina Tag istanza digitare un nome per l'istanza Amazon EC2, quindi scegliere 6. Configura il gruppo di sicurezza.
 - Nella pagina Configura gruppi di sicurezza, seleziona l'opzione Seleziona un gruppo di sicurezza esistente. Per ulteriori informazioni sui gruppi di sicurezza, consulta l'articolo sui [gruppi di sicurezza Amazon EC2 per le istanze Linux](#).



- Scegliere il nome del gruppo di sicurezza di Amazon VPC, quindi scegliere Rivedi e avvia.
- Nella pagina Verifica istanza e avvia, scegliere Avvia.
 - Nella finestra Seleziona una coppia di chiavi esistente o crea una nuova coppia di chiavi, specificare una coppia di chiavi che si desidera utilizzare con questa istanza.

Note

Per informazioni sulla gestione delle coppie di chiavi, consultare la [Guida alle operazioni di base di Amazon EC2](#).

- Quando sei pronto per avviare l'istanza Amazon EC2, scegli Avvia.

È ora possibile assegnare un indirizzo IP elastico all'istanza Amazon EC2 appena creata. È necessario utilizzare questo indirizzo IP per connettersi all'istanza Amazon EC2.

Per assegnare un indirizzo IP elastico (Console)

- Accedere alla console Amazon VPC all'indirizzo <https://console.aws.amazon.com/vpc/>.

2. Nel pannello di navigazione scegliere IP elastici.
3. Scegli Alloca indirizzo IP elastico.
4. Nella finestra di dialogo Alloca indirizzo IP elastico accettare il gruppo di bordi di rete di default e scegliere Alloca .
5. Scegliere l'indirizzo IP elastico appena allocato dall'elenco e scegliere Associa indirizzo.
6. Nella casella di dialogo Associa indirizzo, nella casella Istanza, scegliere l'ID dell'istanza Amazon EC2 che è stata avviata.


Nella casella Indirizzo IP privato scegliere la casella per ottenere l'indirizzo IP privato, quindi scegliere Associa.

È ora possibile utilizzare SSH per connettersi all'istanza Amazon EC2 utilizzando l'indirizzo IP elastico creato.

Eeguire la connessione all'istanza Amazon EC2.

- Aprire una finestra dei comandi. Al prompt dei comandi, eseguire il comando seguente, sostituendo `mykeypair.pem` con il nome del file di coppia di chiavi e `54.207.55.251` con l'indirizzo IP elastico.

```
ssh -i mykeypair.pem ec2-user@54.207.55.251
```

 Important

Non disconnettersi ancora dall'istanza Amazon EC2.

Ora si è pronti per interagire con il cluster di ElastiCache. Prima di farlo, se non lo hai già fatto, devi installare l'utility telnet.

Per installare telnet e interagire con il cluster di cache (AWS CLI)

1. Aprire una finestra dei comandi. Nel prompt dei comandi, eseguire il seguente comando. Alla richiesta di conferma, digitare `y`.

```
sudo yum install telnet  
Loaded plugins: priorities, security, update-motd, upgrade-helper  
Setting up Install Process
```

```
Resolving Dependencies
--> Running transaction check

...(output omitted)...

Total download size: 63 k
Installed size: 109 k
Is this ok [y/N]: y
Downloading Packages:
telnet-0.17-47.7.amzn1.x86_64.rpm                | 63 kB    00:00

...(output omitted)...

Complete!
```

2. Utilizzare telnet per connettersi all'endpoint del nodo di cache sulla porta 6379. Sostituire il nome host mostrato di seguito con il nome host del nodo della cache.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
```

Ora sei connesso al motore della cache e puoi eseguire comandi. In questo esempio, aggiungere un elemento dati alla cache e quindi ottenerlo immediatamente dopo. Infine, disconnettersi dal nodo della cache.

Per memorizzare una chiave e un valore, digitare le due righe seguenti:

```
set mykey myvalue
```

Il motore della cache risponde con quanto segue:

```
OK
```

Per recuperare il valore per `mykey`, digitare quanto segue:

```
get mykey
```

Per disconnettersi dal motore della cache, digitare quanto segue:

```
quit
```

3. Accedere alla console di ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/> e ottenere l'endpoint per uno dei nodi nel cluster di cache. Per maggiori informazioni, consulta [Trovare gli endpoint di connessione](#) per Redis.
4. Utilizzare telnet per connettersi all'endpoint del nodo di cache sulla porta 6379. Sostituire il nome host mostrato di seguito con il nome host del nodo della cache.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
```

Ora sei connesso al motore della cache e puoi eseguire comandi. In questo esempio, aggiungere un elemento dati alla cache e quindi ottenerlo immediatamente dopo. Infine, disconnettersi dal nodo della cache.

Per memorizzare una chiave e un valore, digitare quanto segue:

```
set mykey myvalue
```

Il motore della cache risponde con quanto segue:

```
OK
```

Per recuperare il valore per mykey, digitare quanto segue:

```
get mykey
```

Il motore della cache risponde con quanto segue:

```
get mykey  
myvalue
```

Per disconnettersi dal motore della cache, digitare quanto segue:

```
quit
```

⚠ Important

Per evitare l'addebito di spese aggiuntive sull'account AWS, assicurarsi di eliminare le risorse AWS non più necessarie dopo aver verificato questi esempi.

API Amazon ElastiCache ed endpoint VPC di interfaccia (AWS PrivateLink)

Puoi stabilire una connessione privata tra gli endpoint VPC e dell'API Amazon ElastiCache creando un endpoint VPC di interfaccia. Gli endpoint di interfaccia sono dotati di tecnologia [AWS PrivateLink](#). AWS PrivateLink ti consente di accedere privatamente alle operazioni dell'API Amazon ElastiCache senza un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione AWS Direct Connect.

Le istanze nel VPC non richiedono indirizzi IP pubblici per la comunicazione con gli endpoint dell'API Amazon ElastiCache. Le istanze non richiedono indirizzi IP pubblici neppure per l'uso di una delle operazioni dell'API ElasticCache disponibili. Il traffico di rete tra VPC e Amazon ElastiCache 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 dei VPC, consulta [Endpoint VPC di interfaccia \(AWS PrivateLink\)](#) nella Guida per l'utente di Amazon VPC.
- Per ulteriori informazioni sulle operazioni dell'API ElastiCache, consulta [Operazioni dell'API ElastiCache](#).

Dopo aver creato un endpoint VPC di interfaccia, se abiliti nomi host [DNS privato](#) per l'endpoint, per il VPC viene risolto l'endpoint ElastiCache predefinito (<https://elasticache.Region.amazonaws.com>). Se non abiliti nomi host DNS privati, Amazon VPC fornisce un nome di endpoint DNS che puoi utilizzare nel formato seguente:

```
VPC_Endpoint_ID.elasticache.Region.vpce.amazonaws.com
```

Per ulteriori informazioni, consulta [Endpoint VPC di interfaccia \(AWS PrivateLink\)](#) nella Guida per l'utente di Amazon VPC. ElastiCache supporta l'esecuzione di chiamate a tutte le relative [Operazioni dell'API](#) nel VPC.

Note

I nomi host DNS privati possono essere abilitati per un solo endpoint VPC nel VPC. Per creare un endpoint VPC supplementare, il nome host DNS privato deve essere disabilitato.

Considerazioni sugli endpoint VPC

Prima di configurare un endpoint VPC di interfaccia per gli endpoint dell'API Amazon ElastiCache, rivedi [Proprietà e limitazioni degli endpoint di interfaccia](#) nella Guida per l'utente di Amazon VPC. Tutte le operazioni dell'API ElastiCache pertinenti alla gestione delle risorse di Amazon ElastiCache sono disponibili dal VPC mediante AWS PrivateLink.

Le policy degli endpoint VPC sono supportate per gli endpoint dell'API ElastiCache. Per impostazione predefinita, l'accesso completo alle operazioni dell'API ElastiCache è consentito tramite 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 VPC di interfaccia per l'API ElastiCache

Puoi creare un endpoint VPC per l'API Amazon ElastiCache tramite la console Amazon VPC o AWS 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, puoi abilitare nomi host DNS privati per l'endpoint. Con questa operazione, per l'endpoint VPC viene risolto l'endpoint Amazon ElastiCache predefinito (<https://elasticache.Region.amazonaws.com>). Per le regioni AWS Cina (Pechino) e Cina (Ningxia), puoi creare richieste API con l'endpoint VPC utilizzando elasticache.cn-north-1.amazonaws.com.cn per Pechino e elasticache.cn-northwest-1.amazonaws.com.cn per Ningxia. 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 degli endpoint VPC per l'API Amazon ElastiCache

Puoi allegare una policy degli endpoint all'endpoint VPC che controlla l'accesso all'API ElastiCache. 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 degli endpoint VPC per le operazioni dell'API ElastiCache

Di seguito è riportato un esempio di una policy degli endpoint per l'API ElastiCache. Se collegata a un endpoint, questa policy concede l'accesso alle operazioni dell'API ElastiCache elencate per tutti i principali su tutte le risorse.

```
{
  "Statement": [{
    "Principal": "*",
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:ModifyCacheCluster",
      "elasticache:CreateSnapshot"
    ],
    "Resource": "*"
  }]
}
```

Example Policy degli endpoint VPC che nega tutti gli accessi da un account AWS specificato

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"  
  ]  
}  
]  
}
```

Sottoreti e gruppi di sottoreti

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

Se crei un cluster progettato autonomamente in un Amazon VPC, devi usare un gruppo di sottoreti. ElastiCache utilizza il gruppo di sottoreti per selezionare una sottorete e gli indirizzi IP all'interno di quella sottorete da associare ai nodi.

ElastiCache fornisce un gruppo di sottoreti IPv4 predefinito; in alternativa, puoi scegliere di crearne uno nuovo. Per IPv6 devi creare un gruppo di sottoreti con un blocco CIDR IPv6. Se scegli dual-stack, devi selezionare un tipo di individuazione IP: IPv6 o IPv4.

ElastiCache serverless non utilizza una risorsa gruppo di sottoreti e usa invece un elenco di sottoreti direttamente durante la creazione.

Questa sezione specifica come creare e sfruttare le sottoreti e i gruppi di sottoreti per gestire gli accessi alle risorse di ElastiCache.

Per ulteriori informazioni sull'utilizzo dei gruppi di sottoreti in un ambiente Amazon VPC, consulta [Accesso al cluster o gruppo di replica](#).

Argomenti

- [Creazione di un gruppo di sottoreti](#)
- [Assegnazione di un gruppo di sottoreti ad una cache](#)
- [Modifica di un gruppo di sottoreti](#)
- [Eliminazione di un gruppo di sottoreti](#)

Creazione di un gruppo di sottoreti

Un gruppo di sottoreti di cache è una raccolta di sottoreti che è possibile assegnare alle cache di un VPC. Quando si avvia una cache in un VPC, è necessario scegliere un gruppo di sottoreti di cache. Quindi, ElastiCache utilizza il gruppo di sottoreti di cache per assegnare gli indirizzi IP della sottorete per ogni nodo di cache.

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 liberi, potresti avere delle limitazioni sul numero di nodi che puoi aggiungere a un 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.

Se si sceglie IPV4 come tipo di rete, sarà disponibile un gruppo di sottoreti predefinito; in alternativa, è possibile crearne uno nuovo. ElastiCache utilizza il gruppo di sottoreti per selezionare una sottorete e gli indirizzi IP all'interno di quella sottorete da associare ai nodi. Se si sceglie dual-stack o IPV6, si viene indirizzati alla creazione di sottoreti dual-stack o IPV6. Per ulteriori informazioni sui tipi di rete, consulta [Network type](#). Per ulteriori informazioni, consulta [Creazione di una sottorete nel VPC](#).

Le procedure seguenti illustrano come creare un gruppo di sottoreti chiamato `mysubnetgroup` (console), la AWS CLI e l'API di ElastiCache.

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 gestione di AWS e aprire la console di ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nell'elenco di navigazione scegli Gruppi di sottoreti.
3. Scegli Crea gruppo di sottoreti.
4. Nella procedura guidata Crea gruppo di sottoreti procedi come indicato di seguito. Dopo aver selezionato tutte le impostazioni desiderate, scegli Crea.
 - a. Nella casella Name (Nome), digitare un nome per il gruppo di sottoreti.
 - b. Nella casella Description (Descrizione), digitare una descrizione per il gruppo di sottoreti.
 - c. Nella casella ID VPC seleziona il tuo Amazon VPC.

- d. Per impostazione predefinita sono selezionate tutte le sottoreti. Nel pannello Sottoreti selezionate fai clic su Gestisci e seleziona le zone di disponibilità o le [zone locali](#) e gli ID delle sottoreti private, quindi seleziona Scegli.
5. Nel messaggio di conferma visualizzato, scegliere Close (Chiudi).

Il nuovo gruppo di sottoreti viene visualizzato nell'elenco Subnet Groups (Gruppi di sottoreti) della console di ElastiCache. 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-cache-subnet-group` per creare un gruppo di sottoreti.

Per Linux, macOS o Unix:

```
aws elasticache create-cache-subnet-group \  
  --cache-subnet-group-name mysubnetgroup \  
  --cache-subnet-group-description "Testing" \  
  --subnet-ids subnet-53df9c3a
```

Per Windows:

```
aws elasticache create-cache-subnet-group ^  
  --cache-subnet-group-name mysubnetgroup ^  
  --cache-subnet-group-description "Testing" ^  
  --subnet-ids subnet-53df9c3a
```

Questo comando dovrebbe generare un output simile al seguente:

```
{  
  "CacheSubnetGroup": {  
    "VpcId": "vpc-37c3cd17",  
    "CacheSubnetGroupDescription": "Testing",  
    "Subnets": [  
      {  
        "SubnetIdentifier": "subnet-53df9c3a",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2a"  
        }  
      }  
    ]  
  }  
}
```

```
    }  
  ],  
  "CacheSubnetGroupName": "mysubnetgroup"  
}  
}
```

Per ulteriori informazioni, consulta l'argomento della AWS CLI [create-cache-subnet-group](#).

Assegnazione di un gruppo di sottoreti ad una cache

Dopo aver creato un gruppo di sottoreti, puoi avviare una cache in un Amazon VPC. Per ulteriori informazioni, consulta gli argomenti seguenti.

- Cluster Redis autonomo: per avviare un cluster Redis a nodo singolo, consultare [Creazione di un cluster Redis \(modalità cluster disabilitata\) \(Console\)](#). Nel passaggio 7.a (Advanced Redis Settings (Impostazioni Redis avanzate)), selezionare un gruppo di sottoreti VPC.
- Gruppo di replica Redis (modalità cluster disabilitata): per avviare un gruppo di replica Redis (modalità cluster disabilitata) in un VPC, consultare [Creazione di un gruppo di replica Redis \(modalità cluster disabilitata\) da zero..](#) Nel passaggio 7.b (Advanced Redis Settings (Impostazioni Redis avanzate)), selezionare un gruppo di sottoreti VPC.
- Gruppo di replica Redis (modalità cluster disabilitata) – [Creazione di un cluster Redis \(modalità cluster abilitata\) \(console\)](#).. Nel passaggio 6.i (Advanced Redis Settings (Impostazioni Redis avanzate)), selezionare un gruppo di sottoreti VPC.

Modifica di un gruppo di sottoreti

È possibile modificare una descrizione del gruppo di sottoreti o modificare l'elenco di ID di sottoreti associato al gruppo di sottoreti. Non è possibile eliminare da un gruppo un ID di sottorete se una cache utilizza attualmente quella sottorete.

Le procedure seguenti mostrano come modificare un gruppo di sottoreti.

Modifica dei gruppi di sottoreti (Console)

Per modificare un gruppo di sottoreti

1. Accedere alla AWS Management Console e aprire la console di ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione selezionare Subnet groups (Gruppi di sottoreti).
3. Nell'elenco dei gruppi di sottoreti scegli il pulsante di opzione corrispondente alla sottorete che vuoi modificare e seleziona Modifica.
4. Nel pannello Sottoreti selezionate scegli Gestisci.
5. Apporta eventuali modifiche alle sottoreti selezionate e fai clic su Scegli.
6. Fai clic su Salva le modifiche per salvare le modifiche.

Modifica di gruppi di sottoreti (AWS CLI)

Al prompt dei comandi, utilizzare il comando `modify-cache-subnet-group` per modificare un gruppo di sottoreti.

Per Linux, macOS o Unix:

```
aws elasticache modify-cache-subnet-group \  
  --cache-subnet-group-name mysubnetgroup \  
  --cache-subnet-group-description "New description" \  
  --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Per Windows:

```
aws elasticache modify-cache-subnet-group ^  
  --cache-subnet-group-name mysubnetgroup ^  
  --cache-subnet-group-description "New description" ^  
  --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Questo comando dovrebbe generare un output simile al seguente:

```
{
  "CacheSubnetGroup": {
    "VpcId": "vpc-73cd3c17",
    "CacheSubnetGroupDescription": "New description",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-42dcf93a",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2a"
        }
      },
      {
        "SubnetIdentifier": "subnet-48fc12a9",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2a"
        }
      }
    ],
    "CacheSubnetGroupName": "mysubnetgroup"
  }
}
```

Per ulteriori informazioni, consulta l'argomento della AWS CLI [modify-cache-subnet-group](#).

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 una cache.

Le procedure seguenti mostrano come eliminare un gruppo di sottoreti.

Eliminazione di un gruppo di sottoreti (Console)

Per eliminare un gruppo di sottoreti

1. Accedere alla AWS Management Console e aprire la console di ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione selezionare Subnet groups (Gruppi di sottoreti).
3. Nell'elenco dei gruppi di sottoreti, scegliere quello da eliminare, quindi scegliere Delete (Elimina).
4. Quando ti viene chiesto di confermare questa operazione, digita il nome del gruppo di sottoreti nel campo di immissione testo e scegli Elimina.

Eliminazione di un gruppo di sottoreti (AWS CLI)

Utilizzando l'AWS CLI, chiamare il comando `delete-cache-subnet-group` con il seguente parametro:

- `--cache-subnet-group-name mysubnetgroup`

Per Linux, macOS o Unix:

```
aws elasticache delete-cache-subnet-group \  
  --cache-subnet-group-name mysubnetgroup
```

Per Windows:

```
aws elasticache delete-cache-subnet-group ^  
  --cache-subnet-group-name mysubnetgroup
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta l'argomento della AWS CLI [delete-cache-subnet-group](#).

Identity and Access Management per Amazon ElastiCache

AWS Identity and Access Management (IAM) è un Servizio AWS che consente agli amministratori di controllare in modo sicuro l'accesso alle risorse AWS. Gli amministratori IAM controllano chi può essere autenticato (chi ha effettuato l'accesso) e autorizzato (chi dispone di autorizzazioni) a utilizzare le risorse ElastiCache. IAM è un Servizio AWS il cui uso non comporta costi aggiuntivi.

Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso con policy](#)
- [Funzionamento di Amazon ElastiCache con IAM](#)
- [Esempi di policy basate su identità per Amazon ElastiCache](#)
- [Risoluzione dei problemi relativi all'identità e all'accesso di Amazon ElastiCache](#)
- [Controllo accessi](#)
- [Panoramica della gestione delle autorizzazioni di accesso alle risorse di ElastiCache](#)

Destinatari

Le modalità di utilizzo di AWS Identity and Access Management (IAM) cambiano in base alle operazioni eseguite in ElastiCache.

Utente del servizio: se si utilizza il servizio ElastiCache per eseguire il lavoro, l'amministratore fornisce le credenziali e le autorizzazioni necessarie. All'aumentare del numero più funzionalità ElastiCache utilizzate per eseguire lavoro, potrebbero essere richieste 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 ElastiCache, consultare [Risoluzione dei problemi relativi all'identità e all'accesso di Amazon ElastiCache](#).

Amministratore del servizio: l'utente responsabile delle risorse ElastiCache presso l'azienda dispone dell'accesso completo a ElastiCache. Il suo compito è determinare le caratteristiche e le risorse ElastiCache a cui gli utenti del servizio devono accedere. 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 ulteriori informazioni su come

l'azienda può utilizzare IAM con ElastiCache, consultare [Funzionamento di Amazon ElastiCache con IAM](#).

Amministratore IAM: in qualità di amministratore IAM, potrebbe essere necessario conoscere i dettagli su come scrivere le policy di gestione dell'accesso a ElastiCache. Per visualizzare policy basate su identità di ElastiCache di esempio che puoi utilizzare in IAM, consultare [Esempi di policy basate su identità per Amazon ElastiCache](#).

Autenticazione con identità

L'autenticazione è la procedura di accesso ad AWS con le credenziali di identità. Devi essere autenticato (connesso a AWS) come utente root Utente root dell'account AWS, come utente IAM o assumere un ruolo IAM.

Puoi accedere ad AWS come identità federata utilizzando le credenziali fornite attraverso un'origine di identità. Gli utenti AWS IAM Identity Center (Centro identità IAM), l'autenticazione Single Sign-On (SSO) dell'azienda e le 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. Se accedi ad AWS tramite la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere alla AWS Management Console o al portale di accesso AWS. Per ulteriori informazioni sull'accesso ad AWS, consulta la sezione [Come accedere al tuo Account AWS](#) nella Guida per l'utente di Accedi ad AWS.

Se accedi ad AWS in modo programmatico, AWS fornisce un Software Development Kit (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le richieste utilizzando le tue credenziali. Se non utilizzi gli strumenti AWS, devi firmare le richieste personalmente. Per ulteriori informazioni sulla firma delle richieste, consulta [Firma delle richieste AWS](#) nella Guida per l'utente IAM.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. AWS consiglia ad esempio di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza dell'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 di IAM.

Utente root di un Account AWS

Quando crei un Account AWS, inizi con una singola identità di accesso che ha accesso completo a tutti i Servizi AWS e le risorse nell'account. Tale identità è detta utente root Account AWS ed è

possibile accedervi 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 utilizzarle 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 di IAM.

Identità federata

Come best practice, richiedi agli utenti umani, compresi quelli che richiedono l'accesso di amministratore, di utilizzare la federazione con un provider di identità per accedere a Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory degli utenti aziendali, un provider di identità Web, AWS Directory Service, la directory Identity Center o qualsiasi utente che accede ai Servizi AWS utilizzando le credenziali fornite tramite un'origine di identità. Quando le identità federate accedono agli Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. È possibile creare utenti e gruppi in IAM Identity Center oppure connettersi e sincronizzarsi con un gruppo di utenti e gruppi nell'origine di identità per utilizzarli in tutte le applicazioni e gli Account AWS. Per ulteriori informazioni sul Centro identità IAM, consulta [Cos'è Centro identità IAM?](#) nella Guida per l'utente di AWS IAM Identity Center.

Utenti e gruppi IAM

Un [utente IAM](#) è una identità all'interno del tuo 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, per 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 di 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 Amministratori IAM 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 di IAM.

Ruoli IAM

Un [ruolo IAM](#) è un'identità all'interno di un Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. È possibile assumere temporaneamente un ruolo IAM nella AWS Management Console mediante lo [scambio di ruoli](#). È possibile assumere un ruolo chiamando un'azione AWS CLI o API AWS oppure utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta [Utilizzo di ruoli IAM](#) nella Guida per l'utente di 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 di 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 ulteriori 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, per alcuni dei Servizi AWS, è possibile collegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente di IAM.
- **Accesso multi-servizio:** 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.

- **Inoltro delle sessioni di accesso (FAS):** quando si utilizza un utente o un ruolo IAM per eseguire operazioni in AWS, tale utente o ruolo viene considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra azione in un servizio diverso. FAS utilizza le autorizzazioni del principale che effettua la chiamata a un Servizio AWS, combinate con il Servizio AWS richiedente, per effettuare richieste a servizi a valle. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che necessita di interazioni con altri Servizi AWS o risorse per essere portata a termine. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le operazioni. 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](#) assunto da un servizio per eseguire operazioni per conto dell'utente. 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 di 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 ai servizi sono visualizzati nell'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:** è possibile utilizzare un ruolo IAM per gestire credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 che eseguono richieste di AWS CLI o dell'API AWS. Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un ruolo AWS a un'istanza EC2, affinché sia disponibile per tutte le relative applicazioni, puoi creare un profilo dell'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 di IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta [Quando creare un ruolo IAM \(invece di un utente\)](#) nella Guida per l'utente di IAM.

Gestione dell'accesso con policy

Per controllare l'accesso a AWS è possibile creare policy e collegarle a identità o risorse AWS. Una policy è un oggetto in AWS che, quando associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste policy quando un principale IAM (utente, utente root o sessione ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle policy viene archiviata in AWS sotto forma di 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 di IAM.

Gli amministratori possono utilizzare le policy AWSJSON per specificare l'accesso ai diversi elementi. 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 azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. Successivamente l'amministratore può 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'azione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dalla AWS Management Console, la AWS CLI o l'API AWS.

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 di IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono incorporate direttamente in un singolo utente, gruppo o ruolo. Le policy gestite sono policy autonome che possono essere collegate a più utenti, gruppi e ruoli in Account AWS. Le policy gestite includono le policy gestite da AWS e le policy gestite dal cliente. 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 di IAM.

Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile allegare 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 utilizzarle per controllare l'accesso a una risorsa specifica. Quando è allegata 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 è possibile utilizzare le policy gestite da AWS da IAM in una policy basata su 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 sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3, AWS WAF e Amazon VPC sono esempi di servizi che supportano le ACL. 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 altri tipi di policy 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 funzione 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 di IAM.
- **Policy di controllo dei servizi (SCP):** le SCP sono policy JSON che specificano il numero massimo di autorizzazioni per un'organizzazione o unità organizzativa (OU) in AWS Organizations. AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata degli Account

AWSmultipli 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. La SCP limita le autorizzazioni per le entità negli account membri, compreso ogni 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 di IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per informazioni su come AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle policy](#) nella Guida per l'utente di IAM.

Funzionamento di Amazon ElastiCache con IAM

Prima di utilizzare IAM per gestire l'accesso a ElastiCache, scopri quali funzionalità di IAM sono disponibili per l'uso con ElastiCache.

Funzionalità IAM utilizzabili con Amazon ElastiCache

Funzionalità IAM	Supporto ElastiCache
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione delle policy	Sì

Funzionalità IAM	Supporto ElastiCache
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 ottenere un quadro generale del funzionamento di ElastiCache e altri servizi AWS con la maggior parte delle caratteristiche di IAM, consulta [Servizi AWS supportati da IAM](#) nella Guida per l'utente IAM.

Policy basate su identità per ElastiCache

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 di 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 su identità per ElastiCache

Per visualizzare esempi di policy basate su identità ElastiCache, consultare [Esempi di policy basate su identità per Amazon ElastiCache](#).

Policy basate su risorse all'interno di ElastiCache

Supporta le policy basate su risorse

No

Le policy basate su risorse sono documenti di policy JSON che è possibile allegare 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 utilizzarle per controllare l'accesso a una risorsa specifica. Quando è allegata 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 l'entità principale e la risorsa si trovano in diversi Account AWS, un amministratore IAM nell'account attendibile deve concedere all'entità principale (utente o ruolo) anche l'autorizzazione per 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 [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente di IAM.

Operazioni delle policy per ElastiCache

Supporta le azioni di policy

Sì

Gli amministratori possono utilizzare le policy JSON AWS per specificare gli accessi ai diversi elementi. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni di policy hanno spesso lo stesso nome dell'operazione API AWS. 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 di azioni ElastiCache, consulta [Operazioni definite da Amazon ElastiCache](#) nella Guida di riferimento per l'autorizzazione del servizio.

Le operazioni delle policy in ElastiCache utilizzano il seguente prefisso prima dell'operazione:

```
elasticache
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [  
  "elasticache:action1",  
  "elasticache:action2"  
]
```

È possibile specificare più operazioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le operazioni che iniziano con la parola Describe, includi la seguente operazione:

```
"Action": "elasticache:Describe*"
```

Per visualizzare esempi di policy basate su identità ElastiCache, consultare [Esempi di policy basate su identità per Amazon ElastiCache](#).

Risorse della policy per ElastiCache

Supporta le risorse di policy	Sì
-------------------------------	----

Gli amministratori possono utilizzare le policy JSON AWS per specificare gli accessi ai diversi elementi. 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'azione. 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 di tipi di risorse ElastiCache e i relativi ARN, consulta [Tipi di risorsa definiti da Amazon ElastiCache](#) nella Guida di riferimento per l'autorizzazione del servizio. Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consulta [Operazioni definite da Amazon ElastiCache](#).

Per visualizzare esempi di policy basate su identità ElastiCache, consultare [Esempi di policy basate su identità per Amazon ElastiCache](#).

Chiavi di condizione delle policy per ElastiCache

Supporta le chiavi di condizione delle policy specifiche del servizio	Si
---	----

Gli amministratori possono utilizzare le policy JSON AWS per specificare gli accessi ai diversi elementi. Cioè, quale principale può eseguire operazioni 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 `Condition` è 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 `Condition` in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione AND logica. Se specifichi più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione OR logica. 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 per il servizio. Per visualizzare tutte le chiavi di condizione globali di AWS, consulta [Chiavi di contesto delle condizioni globali di AWS](#) nella Guida per l'utente di IAM.

Per un elenco delle chiavi di condizione di ElastiCache, consulta [Chiavi di condizione per Amazon ElastiCache](#) nella Guida di riferimento per l'autorizzazione del servizio. Per informazioni su operazioni e risorse con cui è possibile utilizzare una chiave di condizione, consulta [Operazioni definite da Amazon ElastiCache](#).

Per visualizzare esempi di policy basate su identità ElastiCache, consultare [Esempi di policy basate su identità per Amazon ElastiCache](#).

Liste di controllo degli accessi (ACL) in ElastiCache

Supporta le ACL

Sì

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni ad 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 su attributi (ABAC) con ElastiCache

Supporta ABAC (tag nelle policy)

Sì

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, tali attributi sono denominati tag. È possibile collegare dei tag alle entità IAM (utenti o ruoli) e a numerose risorse AWS. 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 di 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 Elasticache

Supporta le credenziali temporanee	Sì
------------------------------------	----

Alcuni Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, inclusi i Servizi AWS che funzionano con le credenziali temporanee, consulta [Servizi AWS supportati da IAM](#) nella Guida per l'utente IAM.

Le credenziali temporanee sono utilizzate se si accede alla AWS Management Console utilizzando qualsiasi metodo che non sia la combinazione di nome utente e password. Ad esempio, quando accedi ad AWS utilizzando il collegamento Single Sign-On (SSO) della tua azienda, tale processo crea in automatico 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 di IAM.

È possibile creare manualmente credenziali temporanee utilizzando la AWS CLI o l'API AWS. È quindi possibile utilizzare tali credenziali temporanee per accedere ad AWS. AWS consiglia di generare le credenziali temporanee dinamicamente anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta [Credenziali di sicurezza provvisorie in IAM](#).

Autorizzazioni del principale tra servizi per Elasticache

Supporta sessioni di accesso diretto (FAS)	Sì
--	----

Quando si utilizza un utente o un ruolo IAM per eseguire operazioni in AWS, si viene considerati un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'azione che attiva un'altra azione in un servizio diverso. FAS utilizza le autorizzazioni del principale che effettua la chiamata a un Servizio AWS, combinate con il Servizio AWS richiedente, per effettuare richieste a servizi a valle. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che necessita di interazioni con altri Servizi AWS o risorse per essere portata a termine. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le operazioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina [Forward access sessions](#).

Ruoli di servizio per ElastiCache

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 di IAM.

Warning

La modifica delle autorizzazioni per un ruolo di servizio potrebbe compromettere la funzionalità di ElastiCache. Modificare i ruoli del servizio solo quando ElastiCache fornisce le indicazioni per farlo.

Ruoli collegati ai servizi per ElastiCache

Supporta i ruoli collegati ai servizi	Sì
---------------------------------------	----

Un ruolo collegato ai servizi è un tipo di ruolo di servizio che è collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati ai servizi sono visualizzati nell'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 role (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 su identità per Amazon ElastiCache

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare le risorse ElastiCache. Inoltre, non sono in grado di eseguire attività utilizzando la AWS Management Console, l'AWS Command Line Interface (AWS CLI) o l'API AWS. Per concedere agli

utenti l'autorizzazione per 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 informazioni dettagliate sulle operazioni e sui tipi di risorse definiti da ElastiCache, incluso il formato degli ARN per ogni tipo di risorsa, consultare [Operazioni, risorse e chiavi di condizione per Amazon ElastiCache](#) nella Guida di riferimento per l'autorizzazione del servizio.

Argomenti

- [Best practice per le policy](#)
- [Utilizzo della console ElastiCache](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)

Best practice per le policy

Le policy basate su identità determinano se qualcuno può creare, accedere o eliminare risorse ElastiCache nel tuo account. Queste operazioni possono comportare costi aggiuntivi per l'Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Nozioni di base sulle policy gestite da AWS e passaggio alle autorizzazioni con privilegio minimo: per le informazioni di base su come concedere autorizzazioni a utenti e carichi di lavoro, utilizza le policy gestite da AWS che concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo Account AWS. Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente di AWS 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 privilegi minimi: 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 di 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 inoltre utilizzare le condizioni per concedere l'accesso alle operazioni di servizio, ma solo 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 di 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 di IAM.
- Richiesta dell'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o utenti root nel tuo Account AWS, attiva 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 di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta [Best practice di sicurezza in IAM](#) nella Guida per l'utente di IAM.

Utilizzo della console ElastiCache

Per accedere alla console Amazon ElastiCache, è necessario disporre di un set di autorizzazioni minimo. Queste autorizzazioni devono consentire di elencare e visualizzare i dettagli relativi alle risorse ElastiCache 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 concedere le autorizzazioni minime della console agli utenti che effettuano chiamate solo alla AWS CLI o all'API AWS. Al contrario, concedi l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che gli utenti e i ruoli possano continuare a utilizzare la console ElastiCache, collega anche la policy gestita ConsoleAccess o ReadOnly AWS di ElastiCache 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 allegate alla relativa identità utente. La policy include le

autorizzazioni per completare questa azione sulla console o a livello di programmazione utilizzando la AWS CLI o l'API AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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 di Amazon ElastiCache

Utilizza le informazioni seguenti per diagnosticare e risolvere i problemi comuni che possono verificarsi durante l'utilizzo di ElastiCache e di IAM.

Argomenti

- [Non sono autorizzato/a a eseguire un'operazione in ElastiCache](#)
- [Non sono autorizzato a eseguire iam:PassRole](#)
- [Voglio consentire alle persone esterne al mio account AWS di accedere alle mie risorse ElastiCache](#)

Non sono autorizzato/a a eseguire un'operazione in ElastiCache

Se la AWS Management Console indica che non sei autorizzato a eseguire un'operazione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è la persona che ti ha fornito 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 `elasticache:GetWidget` fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
elasticache:GetWidget on resource: my-example-widget
```

In questo caso, Mateo richiede al suo amministratore di aggiornare le sue policy per poter accedere alla risorsa `my-example-widget` utilizzando l'operazione `elasticache:GetWidget`.

Non sono autorizzato a eseguire iam:PassRole

Se ricevi un errore che indica che non sei autorizzato a eseguire l'operazione `iam:PassRole`, devi aggiornare le policy per poter passare un ruolo a ElastiCache.

Alcuni Servizi AWS consentono di passare un ruolo esistente a tale servizio, invece di creare un nuovo ruolo di servizio o un ruolo collegato ai servizi. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per passare il ruolo al servizio.

L'errore di esempio riportato di seguito si verifica quando un utente IAM denominato `marymajor` tenta di utilizzare la console per eseguire un'operazione in ElastiCache. Tuttavia, l'operazione

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`.

Per ulteriore assistenza con l'accesso, contatta l'amministratore AWS. L'amministratore è colui che ti ha fornito le credenziali di accesso.

Voglio consentire alle persone esterne al mio account AWS di accedere alle mie risorse ElastiCache

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 accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se ElastiCache supporta queste funzionalità, consultare [Funzionamento di Amazon ElastiCache con IAM](#).
- Per informazioni su come garantire l'accesso alle risorse negli Account AWS che possiedi, consulta [Fornire l'accesso a un utente IAM in un altro Account AWS in tuo possesso](#) nella Guida per l'utente di IAM.
- Per informazioni su come fornire l'accesso alle risorse ad Account AWS di terze parti, consulta [Fornire l'accesso agli Account AWS di proprietà di terze parti](#) nella Guida per l'utente di 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 di IAM.
- Per informazioni sulle differenze tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consultare [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente di IAM.

Controllo accessi

Per autenticare le richieste, è necessario disporre di credenziali valide, ma a meno che non si disponga delle autorizzazioni non è possibile creare o accedere a risorse di ElastiCache. Ad esempio, occorre disporre delle autorizzazioni per creare un cluster ElastiCache.

Nelle sezioni seguenti viene descritto come gestire le autorizzazioni per ElastiCache. Consigliamo di leggere prima la panoramica.

- [Panoramica della gestione delle autorizzazioni di accesso alle risorse di ElastiCache](#)
- [Utilizzo di policy basate su identità \(policy IAM\) per Amazon ElastiCache](#)

Panoramica della gestione delle autorizzazioni di accesso alle risorse di ElastiCache

Ogni risorsa AWS è di proprietà di un account AWS e le autorizzazioni necessarie per creare o accedere a una risorsa sono regolate dalle policy 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, Amazon ElastiCache supporta anche il collegamento di policy 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

- [Risorse e operazioni di Amazon ElastiCache](#)

- [Informazioni sulla proprietà delle risorse](#)
- [Gestione dell'accesso alle risorse](#)
- [Policy gestite da AWS per Amazon ElastiCache](#)
- [Utilizzo di policy basate su identità \(policy IAM\) per Amazon ElastiCache](#)
- [Autorizzazioni a livello di risorsa](#)
- [Utilizzo delle chiavi di condizione](#)
- [Utilizzo di ruoli collegati ai servizi per Amazon ElastiCache](#)
- [Autorizzazioni API di ElastiCache: riferimento a operazioni, risorse e condizioni](#)

Risorse e operazioni di Amazon ElastiCache

Per visualizzare un elenco di tipi di risorse ElastiCache e i relativi ARN, consulta [Tipi di risorsa definiti da Amazon ElastiCache](#) nella Guida di riferimento per l'autorizzazione del servizio. Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consulta [Operazioni definite da Amazon ElastiCache](#).

Informazioni sulla proprietà delle risorse

Il proprietario di una risorsa è l'account AWS che crea la risorsa. Ovvero, il proprietario di una risorsa è l'account AWS dell'entità principale che esegue l'autenticazione della richiesta che ha creato la risorsa. Un'entità principale può essere l'account root, un utente IAM o un ruolo IAM. Negli esempi seguenti viene illustrato il funzionamento:

- Supponi di utilizzare le credenziali dell'account root del tuo account AWS per creare un cluster di cache. In questo caso, il tuo account AWS è il proprietario della risorsa. In ElastiCache, la risorsa è il cluster di cache
- Supponi di creare un utente IAM nel tuo account AWS e di concedere a tale utente le autorizzazioni necessarie per creare un cluster di cache. In questo caso, l'utente può creare un cluster di cache. Tieni presente, tuttavia, che l'account AWS a cui appartiene l'utente è il proprietario della risorsa cluster di cache.
- Supponi di creare un ruolo IAM nel tuo account AWS con le autorizzazioni necessarie per creare un cluster di cache. In questo caso, chiunque sia in grado di assumere il ruolo può creare un cluster di cache. L'account AWS a cui appartiene il ruolo è il proprietario della risorsa cluster di cache.

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

In questa sezione viene descritto l'utilizzo di IAM nel contesto di Amazon ElastiCache. 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, le autorizzazioni permettono a tale utente di creare una risorsa ElastiCache, come un cluster di cache, 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 multi-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 multi-account a un altro account AWS (ad esempio l'account B) oppure a un servizio AWS nel modo seguente:
 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 ad assumere tale ruolo a qualsiasi utente dell'account B. In questo modo, gli utenti nell'account B possono creare o accedere alle risorse nell'account A. In alcuni casi potrebbe essere opportuno concedere le autorizzazioni per assumere tale ruolo a un servizio AWS. 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 policy che consente a un utente di eseguire l'operazione `DescribeCacheClusters` per il tuo account AWS. ElastiCache supporta anche l'identificazione di risorse specifiche utilizzando gli ARN delle risorse per le operazioni API. (questo approccio è anche noto come autorizzazioni a livello di risorsa).

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "DescribeCacheClusters",
    "Effect": "Allow",
    "Action": [
      "elasticache:DescribeCacheClusters"],
    "Resource": resource-arn
  ]
}
```

Per ulteriori informazioni sull'uso di policy basate su identità con ElastiCache, consulta [Utilizzo di policy basate su identità \(policy IAM\) per Amazon ElastiCache](#). 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 di Amazon ElastiCache (consulta [Risorse e operazioni di Amazon ElastiCache](#)), il servizio definisce un insieme di operazioni API (consulta [Operazioni](#)). Per concedere le autorizzazioni per queste operazioni API, ElastiCache definisce un set di operazioni che possono essere specificate in una policy. Ad esempio, per la risorsa cluster di ElastiCache, vengono definite le operazioni

`CreateCacheCluster`, `DeleteCacheCluster`, e `DescribeCacheCluster`. 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 [Risorse e operazioni di Amazon ElastiCache](#).
- **Operazione:** utilizzi le parole chiave per identificare le operazioni sulla risorsa da permettere o rifiutare. Ad esempio, a seconda dell'Effect specificato, l'autorizzazione `elasticache:CreateCacheCluster` permette o nega all'utente le autorizzazioni per l'esecuzione dell'operazione di Amazon ElastiCache `CreateCacheCluster`.
- **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 [AWSRiferimento alle policy IAM](#) nella Guida per l'utente di IAM.

Per una tabella che include tutte le operazioni API di Amazon ElastiCache, consulta [Autorizzazioni API di ElastiCache: riferimento a operazioni, 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.

Per esprimere le condizioni è necessario utilizzare chiavi di condizione predefinite. Per usare le chiavi di condizione specifiche di ElastiCache, consulta [Utilizzo delle chiavi di condizione](#). Tuttavia, ci sono

chiavi di condizione ampie AWS disponibili che puoi utilizzare secondo necessità. Per un elenco completo di chiavi AWS, consulta [Chiavi disponibili per le condizioni](#) nella Guida per l'utente di IAM.

Policy gestite da AWS per Amazon ElastiCache

Una policy gestita da AWS è una policy autonoma creata e amministrata da AWS. Le policy gestite da AWS sono progettate per fornire autorizzazioni per molti casi d'uso comuni in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Ricorda che le policy gestite da AWS potrebbero non concedere autorizzazioni con privilegi minimi per i tuoi casi d'uso specifici perché possono essere utilizzate da tutti i clienti AWS. Consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo [policy gestite dal cliente](#) specifiche per i tuoi casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle policy gestite da AWS. Se AWS aggiorna le autorizzazioni definite in una policy gestita da AWS, l'aggiornamento riguarda tutte le identità principali (utenti, gruppi e ruoli) a cui è collegata la policy. È molto probabile che AWS aggiorni una policy gestita da AWS quando viene lanciato un nuovo Servizio AWS o nuove operazioni API diventano disponibili per i servizi esistenti.

Per ulteriori informazioni, consultare [Policy gestite da AWS](#) nella Guida per l'utente di IAM.

Policy gestita da AWS: ElastiCacheServiceRolePolicy

Non è possibile collegare la policy ElastiCacheServiceRolePolicy alle entità IAM. Questa policy è associata a un ruolo collegato ai servizi che consente a ElastiCache di eseguire operazioni per tuo conto.

Questa policy consente a ElastiCache di gestire le risorse AWS per tuo conto, se necessario per la gestione della cache:

- `ec2`: gestisci le risorse di rete EC2 da collegare ai nodi di cache, inclusi endpoint VPC (per cache serverless), interfacce di rete elastica (ENI) (per cluster progettati autonomamente) e gruppi di sicurezza.
- `cloudwatch`: genera i dati delle metriche del servizio in CloudWatch.
- `outposts`: consenti la creazione di nodi di cache su AWS Outposts.

La policy [ElastiCacheServiceRolePolicy](#) è disponibile nella console IAM e in [ElastiCacheServiceRolePolicy](#) nella Guida di riferimento delle policy gestite da AWS.

Policy gestita da AWS: AmazonElastiCacheFullAccess

È possibile allegare la policy AmazonElastiCacheFullAccess alle identità IAM.

Questa policy consente ai principali l'accesso completo a ElastiCache utilizzando la Console di gestione AWS:

- `elasticache`: accedi a tutte le API.
- `iam`: crea il ruolo collegato ai servizi necessario per il funzionamento dei servizi.
- `ec2`: descrivi le risorse EC2 dipendenti necessarie per la creazione della cache (VPC, sottorete, gruppo di sicurezza) e consenti la creazione di endpoint VPC (per cache serverless).
- `kms`: consenti l'utilizzo di CMK gestite dal cliente per la crittografia dei dati a riposo.
- `cloudwatch`: consenti l'accesso alle metriche per visualizzare le metriche di ElastiCache nella console.
- `application-autoscaling`: consenti l'accesso per descrivere le policy di dimensionamento automatico per le cache.
- `logs`: utilizzato per popolare i flussi di log per la funzionalità di distribuzione dei log nella console.
- `firehose`: utilizzato per popolare i flussi di distribuzione per la funzionalità di distribuzione dei log nella console.
- `s3`: utilizzato per popolare i bucket S3 per la funzionalità di ripristino degli snapshot nella console.
- `outposts`: utilizzato per popolare AWS Outposts per la creazione di cache nella console.
- `sns`: utilizzato per popolare gli argomenti SNS per la funzionalità di notifica nella console.

La policy [AmazonElastiCacheFullAccess](#) è disponibile nella console IAM e in [AmazonElastiCacheFullAccess](#) nella Guida di riferimento delle policy gestite da AWS.

Policy gestita da AWS: AmazonElastiCacheReadOnlyAccess

È possibile allegare la policy AmazonElastiCacheReadOnlyAccess alle identità IAM.

Questa policy consente ai principali l'accesso in sola lettura a ElastiCache utilizzando la Console di gestione AWS:

- `elasticache`: accedi alle API `Describe` di sola lettura.

La policy [AmazonElastiCacheReadOnlyAccess](#) è disponibile nella console IAM e in [AmazonElastiCacheReadOnlyAccess](#) nella Guida di riferimento delle policy gestite da AWS.

Aggiornamenti di ElastiCache alle policy gestite da AWS

Visualizza i dettagli sugli aggiornamenti alle policy gestite da AWS per ElastiCache a partire da quando il servizio ha iniziato a monitorare le modifiche. Per gli avvisi automatici sulle modifiche apportate a questa pagina, sottoscrivi il feed RSS nella pagina della cronologia dei documenti di ElastiCache.

Modifica	Descrizione	Data
AmazonElastiCacheFullAccess - Aggiornamento di una policy esistente	ElastiCache ha aggiunto nuove autorizzazioni per consentire la gestione delle cache serverless e l'utilizzo di tutte le funzionalità del servizio tramite la console.	27 novembre 2023
ElastiCacheServiceRolePolicy - Aggiornamento di una policy esistente	ElastiCache ha aggiunto nuove autorizzazioni per consentire la gestione degli endpoint VPC per le risorse di cache serverless.	27 novembre 2023
ElastiCache ha iniziato a monitorare le modifiche	ElastiCache ha iniziato a monitorare le modifiche per le policy gestite da AWS.	7 febbraio 2020

Utilizzo di policy basate su identità (policy IAM) per Amazon ElastiCache

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

In primo luogo, è consigliabile leggere gli argomenti in cui vengono spiegati i concetti di base e le opzioni per gestire l'accesso alle risorse di Amazon ElastiCache. Per ulteriori

informazioni, consulta [Panoramica della gestione delle autorizzazioni di accesso alle risorse di ElastiCache](#).

In questa sezione vengono trattati gli argomenti seguenti:

- [Policy gestite da AWS per Amazon ElastiCache](#)
- [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": [
        "elasticache:CreateServerlessCache",
        "elasticache:CreateCacheCluster",
        "elasticache:DescribeServerlessCaches",
        "elasticache:DescribeReplicationGroups",
        "elasticache:DescribeCacheClusters",
        "elasticache:ModifyServerlessCache",
        "elasticache:ModifyReplicationGroup",
        "elasticache:ModifyCacheCluster"
      ],
      "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 assegna le autorizzazioni per le operazioni di Amazon ElastiCache (`elasticache:Create*`, `elasticache:Describe*`, `elasticache:Modify*`)
- 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 operazioni API di Amazon ElastiCache e le risorse a cui si applicano, consulta [Autorizzazioni API di ElastiCache: riferimento a operazioni, risorse e condizioni](#).

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: assegnazione a un utente dell'autorizzazione a chiamare l'API IAM `CreateServiceLinkedRole`](#)). Oppure è stato creato un ruolo di ElastiCache legato al servizio.

Se combinate con le autorizzazioni minime necessarie per utilizzare la console di Amazon ElastiCache, le policy esemplificative in questa sessione concedono ulteriori autorizzazioni. Tali esempi sono, inoltre, pertinenti alle SDK AWS e alla AWS CLI.

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 operazioni di ElastiCache, apparentemente semplici, in realtà possono necessitare di altre operazioni di supporto, quando si utilizza la console di ElastiCache. Ad esempio, `elasticache:CreateCacheCluster` concede le autorizzazioni necessarie per creare cluster di cache di ElastiCache. Tuttavia, per portare a termine questa operazione, la console di ElastiCache si avvale di una serie di operazioni `Describe` e `List` per compilare gli elenchi di console.

Esempi

- [Esempio 1: assegnazione a un utente dell'accesso in sola lettura alle risorse ElastiCache](#)
- [Esempio 2: assegnazione a un utente dell'autorizzazione a eseguire operazioni di amministrazione comuni del sistema ElastiCache](#)
- [Esempio 3: assegnazione a un utente dell'accesso a tutte le operazioni dell'API ElastiCache](#)
- [Esempio 4: assegnazione a un utente dell'autorizzazione a chiamare l'API IAM `CreateServiceLinkedRole`](#)
- [Esempio 5: assegnazione a un utente dell'autorizzazione a connettersi alla cache serverless utilizzando l'autenticazione IAM](#)

Esempio 1: assegnazione a un utente dell'accesso in sola lettura alle risorse ElastiCache

La policy seguente concede le autorizzazioni necessarie per le operazioni di ElastiCache che permettono all'utente di consultare le risorse in elenco. In genere, si collega questo tipo di policy di autorizzazione a un gruppo di gestori.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "ECReadOnly",
    "Effect": "Allow",
    "Action": [
      "elasticache:Describe*",
      "elasticache:List*"
    ],
    "Resource": "*"
  }
]
```

Esempio 2: assegnazione a un utente dell'autorizzazione a eseguire operazioni di amministrazione comuni del sistema ElastiCache

Le attività comuni dell'amministratore di sistema includono la modifica delle risorse. Un amministratore di sistema potrebbe, inoltre, voler reperire informazioni sugli eventi di ElastiCache. La policy seguente concede a un utente le autorizzazioni necessarie per eseguire le operazioni di ElastiCache correlate alle più comuni attività degli amministratori di sistema, già citate. In genere, si collega questo tipo di policy di autorizzazione al gruppo degli amministratori di sistema.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "ECAAllowMutations",
    "Effect": "Allow",
    "Action": [
      "elasticache:Modify*",
      "elasticache:Describe*",
      "elasticache:ResetCacheParameterGroup"
    ],
    "Resource": "*"
  }
]
```

Esempio 3: assegnazione a un utente dell'accesso a tutte le operazioni dell'API ElastiCache

La policy seguente permette a un utente di accedere a tutte le operazioni di ElastiCache. Consigliamo di concedere questo tipo di policy di autorizzazione solo a un utente amministratore.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "ECAAllowAll",
    "Effect": "Allow",
    "Action": [
      "elasticache:*"
    ],
    "Resource": "*"
  }
]
```

Esempio 4: assegnazione a un utente dell'autorizzazione a chiamare l'API IAM CreateServiceLinkedRole

La policy seguente permette a un utente di chiamare l'API IAM CreateServiceLinkedRole . Consigliamo di concedere questo tipo di policy di autorizzazione all'utente che invoca le operazioni mutative di ElastiCache.

```
{
  "Version": "2012-10-17",
```

```
"Statement":[
  {
    "Sid":"CreateSLRAllows",
    "Effect":"Allow",
    "Action":[
      "iam:CreateServiceLinkedRole"
    ],
    "Resource":"*",
    "Condition":{"
      "StringLike":{"
        "iam:AWSServiceName":"elasticache.amazonaws.com"
      }
    }
  }
]
```

Esempio 5: assegnazione a un utente dell'autorizzazione a connettersi alla cache serverless utilizzando l'autenticazione IAM

La seguente policy consente a un utente di connettersi a una cache serverless utilizzando l'autenticazione IAM tra il 2023-04-01 e il 2023-06-30.

```
{
  "Version" : "2012-10-17",
  "Statement" :
  [
    {
      "Effect" : "Allow",
      "Action" : ["elasticache:Connect"],
      "Resource" : [
        "arn:aws:elasticache:us-east-1:123456789012:serverlesscache:*"
      ],
      "Condition": {
        "DateGreaterThan": {"aws:CurrentTime": "2023-04-01T00:00:00Z"},
        "DateLessThan": {"aws:CurrentTime": "2023-06-30T23:59:59Z"}
      }
    },
    {
      "Effect" : "Allow",
      "Action" : ["elasticache:Connect"],
      "Resource" : [
        "arn:aws:elasticache:us-east-1:123456789012:user:*"
      ]
    }
  ]
}
```

```
    ]
  }
]
}
```

Autorizzazioni a livello di risorsa

È possibile limitare la portata delle autorizzazioni specificando le risorse in una policy IAM. Molte operazioni dell'API ElastiCache supportano un tipo di risorsa che varia a seconda del comportamento dell'operazione. 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.

Per visualizzare un elenco di tipi di risorse ElastiCache e i relativi ARN, consulta [Tipi di risorsa definiti da Amazon ElastiCache](#) nella Guida di riferimento per l'autorizzazione del servizio. Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consulta [Operazioni definite da Amazon ElastiCache](#).

Esempi

- [Esempio 1: consentire a un utente l'accesso completo a tipi di risorse di ElastiCache specifiche](#)
- [Esempio 2: rifiutare a un utente l'accesso a una cache serverless](#)

Esempio 1: consentire a un utente l'accesso completo a tipi di risorse di ElastiCache specifiche

La policy seguente consente esplicitamente tutte le risorse di tipo cache serverless.

```
{
  "Sid": "Example1",
  "Effect": "Allow",
  "Action": "elasticache:*",
  "Resource": [
    "arn:aws:elasticache:us-east-1:account-id:serverlesscache:*"
  ]
}
```

Esempio 2: rifiutare a un utente l'accesso a una cache serverless

Nell'esempio seguente viene negato esplicitamente l'accesso a una determinata cache serverless.

```
{
  "Sid": "Example2",
  "Effect": "Deny",
  "Action": "elasticache:*",
  "Resource": [
    "arn:aws:elasticache:us-east-1:account-id:serverlesscache:name"
  ]
}
```

Utilizzo delle chiavi di condizione

Puoi specificare le condizioni che determinano il modo in cui una policy IAM viene applicata. Su ElastiCache è possibile utilizzare l'elemento `Condition` di una policy JSON per confrontare le chiavi della richiesta con i valori chiave specificati nella policy. Per ulteriori informazioni, consulta [elementi della policy IAM JSON: condizione](#).

Per un elenco delle chiavi di condizione di ElastiCache, consulta [Chiavi di condizione per Amazon ElastiCache](#) nella Guida di riferimento per l'autorizzazione del servizio.

Per un elenco delle chiavi di condizione globali, consulta [Chiavi di contesto delle condizioni globaliAWS](#).

Specifica delle condizioni: Uso delle chiavi di condizione

Per implementare un controllo particolareggiato, scrivi una policy relativa alle autorizzazioni IAM che specifichi le condizioni per controllare un set di singoli parametri su determinate richieste. Quindi la policy viene applicata agli utenti, ai gruppi o ai ruoli IAM creati utilizzando la console IAM.

Per applicare una condizione, aggiungere le informazioni sulla condizione all'istruzione della policy IAM. Nell'esempio seguente viene specificata la condizione secondo la quale qualsiasi cluster di cache progettato autonomamente avrà nodi di tipo `cache.r5.large`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```

    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:parametergroup:*",
      "arn:aws:elasticache:*:*:subnetgroup:*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:cluster:*",
      "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
      "StringEquals": {
        "elasticache:CacheNodeType": [
          "cache.r5.large"
        ]
      }
    }
  }
]
}

```

Per maggiori informazioni, consulta [Esempi di policy di controllo dell'accesso basate su tag](#).

Per ulteriori informazioni sull'utilizzo degli operatori delle condizioni di policy, consulta [Autorizzazioni API di ElastiCache: riferimento a operazioni, risorse e condizioni](#).

Policy di esempio: Utilizzo di condizioni per il controllo granulare dei parametri

In questa sezione vengono illustrate le policy di esempio per implementare il controllo granulare degli accessi sui parametri di ElastiCache elencati in precedenza.

1. `elasticache:MaximumDataStorage`: specifica l'archiviazione di dati massima di una cache serverless. Utilizzando le condizioni fornite, il cliente non può creare cache in grado di archiviare più di una quantità specifica di dati.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDependentResources",
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
        "arn:aws:elasticache:*:*:snapshot:*",
        "arn:aws:elasticache:*:*:usergroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscache:*"
      ],
      "Condition": {
        "NumericLessThanEquals": {
          "elasticache:MaximumDataStorage": "30"
        },
        "StringEquals": {
          "elasticache:DataStorageUnit": "GB"
        }
      }
    }
  ]
}

```

2. `elasticache:MaximumECPUPerSecond`: specifica il valore massimo di ECPU al secondo di una cache serverless. Utilizzando le condizioni fornite, il cliente non può creare cache in grado di eseguire più di un numero specifico di ECPU al secondo.

```

{
  "Version": "2012-10-17",
  "Statement": [

```



```

    {
      "Sid": "AllowDependentResources",
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
        "arn:aws:elasticache:*:*:snapshot:*",
        "arn:aws:elasticache:*:*:usergroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscache:*"
      ],
      "Condition": {
        "NumericLessThanEquals": {
          "elasticache:MaximumECPUPerSecond": "100000"
        }
      }
    }
  ]
}

```

3. `elasticache:CacheNodeType`: Specificare quali `NodeType` un utente può creare. Utilizzando le condizioni fornite, il cliente può specificare un valore singolo o un valore di intervallo per un tipo di nodo.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [

```

```

        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
        "StringEquals": {
            "elasticache:CacheNodeType": [
                "cache.t2.micro",
                "cache.t2.medium"
            ]
        }
    }
}
]
}

```

4. `elasticache:NumNodeGroups`: Creare un gruppo di replica con meno di 20 gruppi di nodi.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        }
    ],
}

```

```

    "Effect": "Allow",
    "Action": [
      "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
      "NumericLessThanEquals": {
        "elasticache:NumNodeGroups": "20"
      }
    }
  }
]
}

```

5. `elasticache:ReplicasPerNodeGroup`: Specificare le repliche per nodo tra 5 e 10.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "NumericGreaterThanEquals": {
          "elasticache:ReplicasPerNodeGroup": "5"
        }
      },
    }
  ]
}

```

```

        "NumericLessThanEquals": {
            "elasticache:ReplicasPerNodeGroup": "10"
        }
    }
}
]
}

```

6. elasticache:EngineVersion: Specificare l'utilizzo della versione del motore 5.0.6.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:EngineVersion": "5.0.6"
        }
      }
    }
  ]
}

```

7. elasticache:EngineType: Specificare l'utilizzo del motore Redis esclusivamente.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:EngineType": "redis"
        }
      }
    }
  ]
}
```

8. elasticache:AtRestEncryptionEnabled: Specificare che i gruppi di replica sarebbero creati solo con la crittografia abilitata.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```

    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "Bool": {
          "elasticache:AtRestEncryptionEnabled": "true"
        }
      }
    }
  ]
}

```

9. elasticache:TransitEncryptionEnabled

- a. Imposta la chiave di condizione `elasticache:TransitEncryptionEnabled` su `false` per l'azione [CreateReplicationGroup](#) per specificare che i gruppi di replica possono essere creati solo quando non viene utilizzato TLS:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",

```

```

        "arn:aws:elasticache:*:*:subnetgroup:*"
    ],
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
        "Bool": {
            "elasticache:TransitEncryptionEnabled": "false"
        }
    }
}
]
}

```

Quando la chiave di condizione `elasticache:TransitEncryptionEnabled` è impostata su `false` in una policy per l'azione [CreateReplicationGroup](#), una richiesta `CreateReplicationGroup` sarà consentita solo se TLS non viene utilizzato (ovvero, se la richiesta non include un parametro `TransitEncryptionEnabled` impostato su `true` o un parametro `TransitEncryptionMode` impostato su `required`).

- b. Imposta la chiave di condizione `elasticache:TransitEncryptionEnabled` su `true` per l'azione [CreateReplicationGroup](#) per specificare che i gruppi di replica possono essere creati solo quando viene utilizzato TLS:

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateReplicationGroup"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        }
    ]
}

```

```

    ],
  },
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
      "Bool": {
        "elasticache:TransitEncryptionEnabled": "true"
      }
    }
  }
]
}

```

Quando la chiave di condizione `elasticache:TransitEncryptionEnabled` è impostata su `true` in una policy per l'azione [CreateReplicationGroup](#), una richiesta `CreateReplicationGroup` sarà consentita solo include un parametro `TransitEncryptionEnabled` impostato su `true` o un parametro `TransitEncryptionMode` impostato su `required`.

- c. Imposta `elasticache:TransitEncryptionEnabled` su `true` per l'azione `ModifyReplicationGroup` per specificare che i gruppi di replica possono essere modificati solo quando viene utilizzato TLS:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:ModifyReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {

```



```

        "BoolIfExists": {
            "elasticache:TransitEncryptionEnabled": "true"
        }
    }
}
]
}

```

Quando la chiave di condizione `elasticache:TransitEncryptionEnabled` è impostata su `true` in una policy per l'azione [ModifyReplicationGroup](#), una richiesta `ModifyReplicationGroup` sarà consentita solo include un parametro `TransitEncryptionMode` impostato su `required`. Facoltativamente, è anche possibile includere il parametro `TransitEncryptionEnabled` impostato su `true`, ma in questo caso non è necessario abilitare TLS.

10`elasticache:AutomaticFailoverEnabled`: Specificare che i gruppi di replica sarebbero creati solo con il failover automatico abilitato.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "Bool": {

```

```

        "elasticache:AutomaticFailoverEnabled": "true"
    }
}
]
}

```

11 `elasticache:MultiAZEnabled`: Specificare che i gruppi di replica non possono essere creati con Multi-AZ disabilitato.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "Bool": {
          "elasticache:MultiAZEnabled": "false"
        }
      }
    }
  ]
}

```

12.`elasticache:ClusterModeEnabled`: Specificare che i gruppi di replica possono essere creati solo con la modalità cluster abilitata.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "Bool": {
          "elasticache:ClusterModeEnabled": "true"
        }
      }
    }
  ]
}
```

13.`elasticache:AuthTokenEnabled`: Specificare che i gruppi di replica possono essere creati solo con il token AUTH abilitato.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```

    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:parametergroup:*",
      "arn:aws:elasticache:*:*:subnetgroup:*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:cluster:*",
      "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
      "Bool": {
        "elasticache:AuthTokenEnabled": "true"
      }
    }
  }
]
}

```

14 `elasticache:SnapshotRetentionLimit`: Specificare il numero di giorni (oppure minimo/massimo) concessi per conservare lo snapshot. Di seguito la policy impone l'archiviazione dei backup per almeno 30 giorni.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
    }
  ]
}

```

```

    "Resource": [
      "arn:aws:elasticache:*:*:parametergroup:*",
      "arn:aws:elasticache:*:*:subnetgroup:*"
    ],
  },
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:CreateReplicationGroup",
      "elasticache:CreateServerlessCache"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:cluster:*",
      "arn:aws:elasticache:*:*:replicationgroup:*",
      "arn:aws:elasticache:*:*:serverlesscache:*"
    ],
    "Condition": {
      "NumericGreaterThanEquals": {
        "elasticache:SnapshotRetentionLimit": "30"
      }
    }
  }
]
}

```

15 `elasticache:KmsKeyId`: Specificare l'utilizzo delle chiavi KMS AWS gestite dal cliente.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDependentResources",
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
        "arn:aws:elasticache:*:*:snapshot:*",
        "arn:aws:elasticache:*:*:usergroup:*"
      ]
    }
  ],
}

```

```

{
  "Effect": "Allow",
  "Action": [
    "elasticache:CreateServerlessCache"
  ],
  "Resource": [
    "arn:aws:elasticache:*:*:serverlesscache:*"
  ],
  "Condition": {
    "StringEquals": {
      "elasticache:KmsKeyId": "my-key"
    }
  }
}
]
}

```

16 `elasticache:CacheParameterGroupName`: Specificare un gruppo di parametri non predefinito con parametri specifici di un'organizzazione sui tuoi cluster. È inoltre possibile specificare un modello di denominazione per i gruppi di parametri o eliminare blocchi su un nome di gruppo di parametri specifico. Di seguito è riportato un esempio che vincola l'utilizzo di solo "my-org-param-group".

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ]
    }
  ]
}

```

```

    ],
    "Resource": [
      "arn:aws:elasticache:*:*:cluster:*",
      "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
      "StringEquals": {
        "elasticache:CacheParameterGroupName": "my-org-param-group"
      }
    }
  }
]
}

```

17. `elasticache:CreateCacheCluster`: rifiuto dell'azione `CreateCacheCluster` se il tag della richiesta `Project` manca o è diverso da `Dev`, `QA` o `Prod`.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*",
        "arn:aws:elasticache:*:*:securitygroup:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "Null": {
          "aws:RequestTag/Project": "true"
        }
      }
    }
  ]
}

```

```

    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "elasticache:CreateCacheCluster",
    "elasticache:AddTagsToResource"
  ],
  "Resource": "arn:aws:elasticache:*:*:cluster:*",
  "Condition": {
    "StringEquals": {
      "aws:RequestTag/Project": [
        "Dev",
        "Prod",
        "QA"
      ]
    }
  }
}
]
}

```

18 `elasticache:CacheNodeType`: consente `CreateCacheCluster` con `cacheNodeType` `cache.r5.large` o `cache.r6g.4xlarge` e tag `Project=XYZ`.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [

```



```
    "elasticache:CreateCacheCluster"
  ],
  "Resource": [
    "arn:aws:elasticache:*:*:cluster:*"
  ],
  "Condition": {
    "StringEqualsIfExists": {
      "elasticache:CacheNodeType": [
        "cache.r5.large",
        "cache.r6g.4xlarge"
      ]
    },
    "StringEquals": {
      "aws:RequestTag/Project": "XYZ"
    }
  }
}
```

Note

Quando si creano policy per applicare tag e altre chiavi di condizione insieme, il condizionale `IfExists` può essere richiesto su elementi di condizione chiave a causa dei requisiti della policy `elasticache:AddTagsToResource` aggiuntivi per le richieste di creazione con il parametro `--tags`.

Utilizzo di ruoli collegati ai servizi per Amazon ElastiCache

Amazon ElastiCache utilizza [ruoli collegati ai servizi](#) di AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo univoco di ruolo IAM collegato direttamente a un servizio AWS, come Amazon ElastiCache. I ruoli collegati al servizio di Amazon ElastiCache sono predefiniti da Amazon ElastiCache. Includono tutte le autorizzazioni necessarie al servizio per richiamare servizi AWS per conto dei cluster.

Un ruolo collegato ai servizi semplifica la configurazione di Amazon ElastiCache perché permette di evitare l'aggiunta manuale delle autorizzazioni necessarie. I ruoli sono già presenti nel tuo account AWS ma sono collegati ai casi d'uso di Amazon ElastiCache e dispongono di autorizzazioni predefinite. Solo Amazon ElastiCache può assumere questi ruoli e solo questi ruoli possono usare

la policy delle autorizzazioni predefinite. È possibile eliminare i ruoli solo dopo aver eliminato le risorse correlate. Questa procedura protegge le risorse di Amazon ElastiCache poiché impedisce la rimozione involontaria delle necessarie autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta [Servizi AWS che funzionano con IAM](#) e cerca i servizi che riportano Sì nella colonna Ruolo associato ai servizi. Scegliere un link Yes (Sì) per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Indice

- [Autorizzazioni del ruolo collegato ai servizi per Amazon ElastiCache](#)
 - [Autorizzazioni per creare un ruolo collegato ai servizi](#)
- [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 ElastiCache](#)
 - [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 ElastiCache](#)
 - [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 del ruolo collegato ai servizi per Amazon ElastiCache

Autorizzazioni per creare un ruolo collegato ai servizi

Per consentire a un'entità IAM di creare il ruolo collegato ai servizi AWSServiceRoleForElastiCache

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/elasticache.amazonaws.com/
AWSServiceRoleForElastiCache*",
    "Condition": {"StringLike": {"iam:AWSServiceName": "elasticache.amazonaws.com"}}
  }

```

Per consentire a un'entità IAM di eliminare il ruolo collegato ai servizi
AWSServiceRoleForElastiCache

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/elasticache.amazonaws.com/
AWSServiceRoleForElastiCache*",
  "Condition": {"StringLike": {"iam:AWSServiceName": "elasticache.amazonaws.com"}}
}

```

In alternativa, è possibile utilizzare una policy gestita da AWS per fornire l'accesso completo ad Amazon ElastiCache.

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 alla AWS Management Console e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione della console IAM seleziona Roles (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 **Or select a service to view its use cases (Oppure seleziona un servizio per visualizzarne i casi d'uso)**, scegli **ElastiCache**.
5. Scegli **Successivo: Autorizzazioni**.
6. In **Policy name (Nome policy)**, si noti che **ElastiCacheServiceRolePolicy** 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)

È possibile utilizzare le operazioni di IAM da AWS Command Line Interface per creare un ruolo collegato ai servizi. 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 elasticache.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 `elasticache.amazonaws.com`.

Modifica della descrizione di un ruolo collegato ai servizi per Amazon ElastiCache

Amazon ElastiCache non consente di modificare il ruolo collegato ai servizi `AWSServiceRoleForElastiCache`. 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 pannello di navigazione della console IAM seleziona Roles (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)

È possibile utilizzare le operazioni di IAM dalla AWS Command Line Interface per modificare la descrizione di un ruolo collegato ai servizi.

Per modificare la descrizione di un ruolo collegato ai servizi (CLI)

1. (Facoltativo) Per visualizzare la descrizione corrente di un ruolo, utilizzare la AWS CLI per l'operazione di IAM [get-role](#).

Example

```
$ aws iam get-role --role-name AWSServiceRoleForElastiCache
```

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 ai servizi, utilizza la AWS CLI per l'operazione di IAM [update-role-description](#).

Per Linux, macOS o Unix:

```
$ aws iam update-role-description \  
  --role-name AWSServiceRoleForElastiCache \  
  --description "new description"
```

Per Windows:

```
$ aws iam update-role-description ^  
  --role-name AWSServiceRoleForElastiCache ^  
  --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=AWSServiceRoleForElastiCache  
&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=AWSServiceRoleForElastiCache  
&Version=2010-05-08  
&Description="New description"
```

Eliminazione di un ruolo collegato ai servizi per Amazon ElastiCache

Se non è più necessario utilizzare una caratteristica o un servizio che richiede un ruolo collegato ai servizi, 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 ElastiCache non elimina il ruolo collegato ai servizi per tuo conto.

Pulizia di un ruolo collegato ai servizi

Prima di utilizzare IAM per eliminare un ruolo collegato ai servizi, verifica innanzitutto che il ruolo non abbia risorse (cluster o gruppi di replica) associati al ruolo.

Per verificare se il ruolo collegato ai servizi dispone di una sessione attiva nella console IAM

1. Accedi alla AWS Management Console e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione della console IAM seleziona Roles (Ruoli). Quindi, scegli il nome (non la casella di controllo) del ruolo AWSServiceRoleForElastiCache
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 Amazon ElastiCache che richiedono AWSServiceRoleForElastiCache

- Per eliminare un cluster, consultare i seguenti argomenti:
 - [Utilizzo di AWS Management Console](#)
 - [Utilizzo di AWS CLI](#)
 - [Utilizzo dell'API ElastiCache](#)
- Per eliminare un gruppo di replica, consultare i seguenti argomenti:
 - [Eliminazione di un gruppo di replica \(console\)](#)
 - [Eliminazione di un gruppo di replica \(AWS CLI\)](#)
 - [Eliminazione di un gruppo di replica \(API ElastiCache\)](#)

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 alla AWS Management Console e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione della console IAM seleziona Roles (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 finestra di dialogo di conferma controlla i dati relativi all'ultimo accesso ai servizi, che indicano quando ognuno dei ruoli selezionati ha effettuato l'accesso a un servizio AWS. 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)

È possibile utilizzare le operazioni di IAM dalla AWS Command Line Interface per eliminare un ruolo collegato ai servizi.

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-`

iddalla 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.

Autorizzazioni API di ElastiCache: riferimento a operazioni, risorse e condizioni

Utilizzare la seguente tabella come riferimento quando si configura il [controllo dell'accesso](#) e si scrivono policy di autorizzazione da collegare a policy IAM (policy basate sulle identità o sulle risorse). La tabella seguente elenca le operazioni API di Amazon ElastiCache e le operazioni corrispondenti cui hai concesso le autorizzazioni di esecuzione dell'operazione. 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 (*).

Puoi usare le chiavi di condizione nelle policy di ElastiCache per esprimere le condizioni. Per visualizzare un elenco delle chiavi di condizione specifiche di Elasticache, insieme alle operazioni e ai tipi di risorse a cui si applicano, consulta [Utilizzo delle chiavi di condizione](#). Per l'elenco completo delle chiavi AWS, consulta [Chiavi di contesto e condizione globali di AWS](#) nella Guida per l'utente di IAM.

Note

Per specificare un'operazione, utilizza il prefisso `elasticache:` seguito dal nome dell'operazione API (ad esempio, `elasticache:DescribeCacheClusters`).

Per visualizzare un elenco di azioni ElastiCache, consulta [Operazioni definite da Amazon ElastiCache](#) nella Guida di riferimento per l'autorizzazione del servizio.

Convalida della conformità per Amazon ElastiCache


Revisori di terza parte valutano la sicurezza e la conformità di AWS come parte di diversi programmi di conformità di AWS, come SOC, PCI, FedRAMP, HIPAA e altri.

Per sapere se il Servizio AWS è coperto da programmi di conformità specifici, consulta i [Servizi AWS coperti dal programma di conformità](#) e scegli il programma di conformità desiderato. Per informazioni generali, consulta [Programmi per la conformità di AWS](#).

È possibile scaricare i report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Download di report in AWS Artifact](#).

La responsabilità di conformità durante l'utilizzo dei Servizi AWS è determinata dalla riservatezza dei dati, dagli obiettivi di conformità dell'azienda e dalle normative vigenti. Per semplificare il rispetto della conformità, AWS mette a disposizione le seguenti risorse:

- [Guide Quick Start per la sicurezza e conformità](#): queste guide all'implementazione illustrano considerazioni relative all'architettura e forniscono la procedura per l'implementazione di ambienti di base su AWS incentrati sulla sicurezza e sulla conformità.
- [Architetture per la sicurezza e la conformità HIPAA su Amazon Web Services](#): questo whitepaper descrive come le aziende possono utilizzare AWS per creare applicazioni conformi alla normativa HIPAA.

 Note

Non tutti i Servizi AWS sono conformi ai requisiti HIPAA. Per ulteriori informazioni, consulta la sezione [Riferimenti sui servizi conformi ai requisiti HIPAA](#).

- [Risorse per la conformità AWS](#): una raccolta di cartelle di lavoro e guide suddivise per settore e area geografica.
- [AWS Guide alla conformità dei clienti](#): comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization (ISO)).
- [Valutazione delle risorse con le regole](#) nella Guida per gli sviluppatori di AWS Config: il servizio AWS Config valuta il livello di conformità delle configurazioni delle risorse con pratiche interne, linee guida e regolamenti.
- [AWS Security Hub](#): questo Servizio AWS fornisce una visione completa dello stato di sicurezza all'interno di AWS. La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina [Documentazione di riferimento sui controlli della Centrale di sicurezza](#).
- [AWS Audit Manager](#): questo Servizio AWS aiuta a verificare continuamente l'utilizzo di AWS per semplificare la gestione dei rischi e della conformità alle normative e agli standard di settore.

Ulteriori informazioni

Per informazioni generali sulla conformità a Cloud AWS, consulta i seguenti documenti:

- [Endpoint FIPS per servizio](#)
- [Aggiornamenti del servizio in ElastiCache](#)
- [Conformità di sicurezza nel cloud AWS](#)
- [Modello di responsabilità condivisa](#)
- [AWSProgramma di Conformità PCI DSS](#)

Resilienza in Amazon ElastiCache

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 all'infrastruttura globale AWS, Amazon ElastiCache offre numerose caratteristiche per supportare la resilienza dei dati e le esigenze di backup.

Argomenti

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

Limitazione dell'impatto degli errori

Quando pianifichi l' ElastiCache implementazione di Amazon, devi pianificare in modo 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.

Argomenti

- [Limitazione degli errori con Redis in esecuzione](#)

- [Raccomandazioni](#)

Limitazione degli errori con Redis in esecuzione

Per minimizzare l'impatto degli errori di nodi e zone di disponibilità con un motore Redis in esecuzione, è possibile avvalersi delle seguenti soluzioni.

Limitazione degli errori dei nodi

Le cache serverless mitigano automaticamente gli errori dei nodi con un'architettura multi-AZ in modo che gli errori dei nodi siano trasparenti per l'applicazione. I cluster progettati autonomamente devono essere configurati in modo appropriato per mitigare l'errore di un singolo nodo.

Per minimizzare l'impatto degli errori dei nodi Redis nei cluster progettati autonomamente, è possibile avvalersi delle seguenti opzioni:

Argomenti

- [Limitazione degli errori con i gruppi di replica Redis](#)

Limitazione degli errori con i gruppi di replica Redis

Un gruppo di replica Redis 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. Quanto scritto sul nodo primario si propaga in modo asincrono ai nodi della replica di lettura.

Quando una replica di lettura genera un errore

1. ElastiCache rileva la replica di lettura non riuscita.
2. ElastiCache disconnette il nodo guasto.
3. ElastiCache avvia e fornisce un nodo sostitutivo nella stessa AZ.
4. Il nuovo nodo si sincronizza con il nodo primario.

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

Redis Multi-AZ

È possibile abilitare la funzione Multi-AZ nei gruppi di replica Redis. Indipendentemente dal fatto che la funzione Multi-AZ sia abilitata o meno, un nodo primario non riuscito sarà comunque rilevato e sostituito automaticamente, sebbene con modalità diverse in base all'abilitazione.

Quando la funzione Multi-AZ è abilitata

1. ElastiCache rileva il guasto del nodo principale.
2. ElastiCache promuove il nodo di replica di lettura con il minor ritardo di replica sul nodo primario.
3. Le altre repliche si sincronizzano con il nuovo nodo primario.
4. ElastiCache avvia una replica di lettura nella zona di disponibilità del sistema primario fallito.
5. Il nuovo nodo si sincronizza con il nuovo nodo primario.

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 velocemente con la funzione Multi-AZ abilitata.

Per ulteriori informazioni, consulta [Riduzione dei tempi di inattività in ElastiCache for Redis con Multi-AZ](#).

Quando la funzione Multi-AZ è disabilitata

1. ElastiCache rileva il guasto primario.
2. ElastiCache mette offline il principale.
3. ElastiCache crea e fornisce un nuovo nodo primario per sostituire il nodo primario guasto.
4. ElastiCache sincronizza il nuovo primario con una delle repliche esistenti.
5. A sincronizzazione conclusa, il nuovo nodo funge da nodo primario del cluster.

Durante le fasi 1-4 di questo processo, l'applicazione non può scrivere sul nodo primario, ma può continuare a leggere dai nodi di replica.

Per precauzione, consigliamo di avviare i nodi nel gruppo di replica avvalendosi di più zone di disponibilità (AZ). Così facendo, l'eventuale malfunzionamento di una zona di disponibilità andrà a condizionare solo i nodi in essa, senza compromettere quelli nelle altre.

Per ulteriori informazioni, consulta [Alta disponibilità utilizzando gruppi di replica](#).

Limitazione degli errori delle zone di disponibilità

Le cache serverless mitigano automaticamente gli errori delle zone di disponibilità con un'architettura multi-AZ replicata in modo che gli errori delle zone di disponibilità siano trasparenti per l'applicazione.

Per limitare l'impatto dell'errore di una zona di disponibilità in un cluster progettato autonomamente, posiziona i nodi di ogni partizione nel maggior numero possibile di zone.

Se tutti i nodi di una partizione, indipendentemente dal numero, fossero posizionati nella stessa zona di disponibilità, un errore di quest'ultima si rivelerebbe catastrofico, poiché comporterebbe la perdita di tutti i dati della partizione. Se si collocano i nodi in più zone, invece, il guasto di una di esse comporta la perdita solo dei nodi ivi contenuti.

Perdere un nodo significa subire un calo nelle prestazioni, poiché le operazioni di lettura si ritrovano a dover essere condivise da meno nodi. Tale calo non conoscerà soluzione di continuità fino alla sostituzione dei nodi.

Per informazioni sulla specifica delle zone di disponibilità per i nodi Redis, consulta [Creazione di un cluster Redis \(modalità cluster disabilitata\) \(Console\)](#).

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

Raccomandazioni

Consigliamo di creare le cache serverless su cluster progettati autonomamente, in quanto si ottiene automaticamente una migliore tolleranza agli errori senza configurazioni aggiuntive. Quando si crea un cluster progettato autonomamente, tuttavia, è necessario pianificare due tipi di errori: errori di singoli nodi ed errori di zone di disponibilità. Il piano di mitigazione degli errori ideale consente di affrontare entrambe le tipologie di guasto.

Minimizzazione dell'impatto degli errori dei nodi

Per ridurre al minimo l'impatto degli errori di nodo, consigliamo l'implementazione con più nodi in ogni shard e la distribuzione dei nodi tra più zone di disponibilità. Questa operazione viene eseguita automaticamente per le cache serverless.

Per i cluster progettati autonomamente, si consiglia di abilitare Multi-AZ sul gruppo di replica in modo che esegua ElastiCache automaticamente il failover su una replica in caso di guasto del nodo primario.

Minimizzare l'impatto degli errori delle zone di disponibilità

Per ridurre al minimo l'impatto dell'errore di una zona di disponibilità, consigliamo di ubicare i nodi nel maggior numero possibile di zone di disponibilità. Distribuire i nodi uniformemente tra le zone

di disponibilità ridurrà notevolmente l'impatto di un eventuale, ancorché improbabile, guasto di una zona. Questa operazione viene eseguita automaticamente per le cache serverless.

Altre precauzioni

A chi esegue Redis, oltre a quanto suddetto, consigliamo di prevedere backup regolari del cluster. I backup (gli snapshot) generano un file .rdb utile per ripristinare la cache in caso di errore o danneggiamento. Per ulteriori informazioni, consulta [Snapshot e ripristino](#).

Sicurezza dell'infrastruttura in AWS ElastiCache

Come servizio gestito, AWS ElastiCache è protetto dalle procedure di sicurezza di rete globali di AWS descritte nella sezione Sicurezza e conformità nel [Centro di architettura AWS](#).

Utilizza le chiamate API pubblicate da AWS per accedere a ElastiCache tramite la rete. I client devono supportare Transport Layer Security (TLS) 1.2 o versioni successive. È consigliabile TLS 1.3 o versioni successive. 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 utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. In alternativa, è possibile utilizzare [AWS Security Token Service](#) (AWS STS) per generare le credenziali di sicurezza temporanee per sottoscrivere le richieste.

Aggiornamenti del servizio in ElastiCache

ElastiCache monitora automaticamente la tua flotta di cache, cluster e nodi per applicare gli aggiornamenti del servizio non appena sono disponibili. Gli aggiornamenti di servizio per le cache serverless vengono applicati automaticamente e in modo trasparente. Per i cluster progettati autonomamente, è necessario impostare una finestra di manutenzione predefinita in modo da poter applicare questi aggiornamenti. ElastiCache Tuttavia, in alcuni casi questo approccio potrebbe risultare troppo rigido e vincolare per i flussi aziendali.

Con gli aggiornamenti di servizio controlli quando e quali aggiornamenti vengono applicati ai cluster progettati autonomamente. È inoltre possibile monitorare lo stato di avanzamento di questi aggiornamenti nel ElastiCache cluster selezionato in tempo reale.

Gestione degli aggiornamenti del servizio

ElastiCache gli aggiornamenti dei servizi per i cluster progettati autonomamente vengono rilasciati regolarmente. Se disponi di uno o più cluster qualificati autoprogettati per tali aggiornamenti del servizio, ricevi notifiche tramite e-mail, SNS, Personal Health Dashboard (PHD) ed CloudWatch eventi Amazon quando gli aggiornamenti vengono rilasciati. Gli aggiornamenti vengono visualizzati anche nella pagina Service Updates sulla console. ElastiCache Utilizzando questa dashboard, puoi visualizzare tutti gli aggiornamenti del servizio e il relativo stato per il tuo ElastiCache parco veicoli. Gli aggiornamenti di servizio per le cache serverless vengono applicati in modo trasparente e non possono essere gestiti tramite la pagina Aggiornamenti del servizio.

Si controlla quando applicare un aggiornamento prima dell'avvio dell'aggiornamento automatico. Ti consigliamo vivamente di applicare qualsiasi aggiornamento di tipo security-update il prima possibile per garantire che i tuoi ElastiCache cluster siano sempre up-to-date dotati delle patch di sicurezza correnti.

Le seguenti sezioni esplorano queste opzioni in dettaglio:

Argomenti

- [Applicazione degli aggiornamenti di servizio](#)
- [Verifica dell'applicazione dell'ultimo aggiornamento del servizio tramite la console AWS](#)
- [Interruzione degli aggiornamenti del servizio](#)

Applicazione degli aggiornamenti di servizio

Puoi iniziare ad applicare gli aggiornamenti di servizio al parco istanze dal momento in cui lo stato degli aggiornamenti è available. Gli aggiornamenti di servizio sono cumulativi. In altre parole, tutti gli aggiornamenti non ancora applicati sono inclusi con l'ultimo aggiornamento.

Se per un aggiornamento del servizio è abilitato l'aggiornamento automatico, puoi scegliere di non intraprendere alcuna azione quando diventa disponibile. ElastiCache pianificherà l'applicazione dell'aggiornamento durante una delle prossime finestre di manutenzione dei cluster dopo la data di inizio dell'aggiornamento automatico. Riceverai notifiche correlate per ogni fase dell'aggiornamento.

Note

Puoi applicare solo aggiornamenti di servizio con stato availableoscheduled.

Per ulteriori informazioni sulla revisione e l'applicazione di eventuali aggiornamenti specifici del servizio ai cluster applicabili ElastiCache, consulta [Applicazione degli aggiornamenti del servizio tramite la console](#)

Quando è disponibile un nuovo aggiornamento del servizio per uno o più ElastiCache cluster, puoi utilizzare la ElastiCache console, l'API o AWS CLI 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 di servizio disponibili, assieme ad altre informazioni, accedi alla pagina Aggiornamenti di servizio nella console.

1. Accedi AWS Management Console e apri la ElastiCache console Amazon all'[indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Nel riquadro di navigazione, scegli Aggiornamenti di servizio.
3. Sotto Aggiornamenti di servizio puoi visualizzare quanto segue:
 - Nome aggiornamento di servizio: il nome univoco dell'aggiornamento di servizio
 - Tipo di aggiornamento: il tipo di aggiornamento di servizio, cioè security-update o engine-update
 - Gravità aggiornamento): la priorità di applicazione dell'aggiornamento:
 - critical (critica): si consiglia di applicare immediatamente questo aggiornamento (entro 14 giorni o meno).
 - important (importante): si consiglia di applicare questo aggiornamento non appena l'attività aziendale lo consente (entro 30 giorni o meno).
 - medium (media): si consiglia di applicare questo aggiornamento il prima possibile (entro 60 giorni o meno).
 - low (bassa): si consiglia di applicare questo aggiornamento il prima possibile (entro 90 giorni o meno).
 - Versione del motore: se il tipo di aggiornamento è aggiornamento del motore, è la versione del motore che viene aggiornata.
 - Data di rilascio: indica quando l'aggiornamento è stato rilasciato per essere applicato sui cluster.
 - Data limite consigliata: data ElastiCache indicativa entro la quale applicare gli aggiornamenti.
 - Status (Stato): lo stato dell'aggiornamento, che è uno dei seguenti:

- **disponibile:** l'aggiornamento è disponibile per i cluster necessari.
 - **complete:** l'aggiornamento è stato applicato.
 - **cancelled (annullato):** l'aggiornamento è stato annullato e non è più necessario.
 - **expired (scaduto):** l'aggiornamento non è più disponibile per l'applicazione.
4. Scegli un aggiornamento singolo (non il pulsante a sinistra) per visualizzare i dettagli dell'aggiornamento di servizio.

Nella sezione Stato aggiornamento cluster puoi visualizzare un elenco di cluster in cui l'aggiornamento di servizio non è stato applicato o è stato applicato recentemente. Per ogni cluster puoi visualizzare:

- **Nome cluster:** il nome del cluster
- **Nodi aggiornati:** il rapporto tra i singoli nodi in un cluster specifico che sono stati aggiornati o rimangono disponibili per l'aggiornamento di servizio specifico.
- **Tipo di aggiornamento:** il tipo di aggiornamento di servizio, cioè security-update o engine-update
- **Stato:** lo stato dell'aggiornamento di servizio sul cluster, cioè:
 - **available:** l'aggiornamento è disponibile per i cluster richiesti.
 - **in corso:** l'aggiornamento del cluster è in fase di applicazione.
 - **scheduled (pianificato):** la data di aggiornamento è stata pianificata.
 - **complete (completo):** 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 stato **available** o **scheduled** e hai selezionato **Applica 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 elasticache describe-service-updates --service-update-status
available
```

Per ulteriori informazioni, vedere [describe-service-updates](#).

- Per applicare un aggiornamento di servizio a un elenco di cluster, utilizza il comando seguente:

```
aws elasticache batch-apply-update-action --service-update
ServiceUpdateNameToApply=sample-service-update --cluster-names cluster-1
cluster2
```

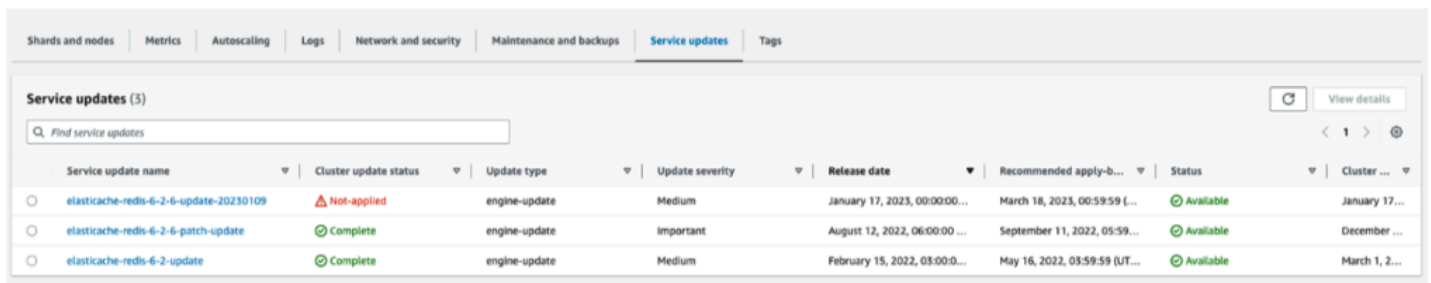
Per ulteriori informazioni, vedere [batch-apply-update-action](#).

Verifica dell'applicazione dell'ultimo aggiornamento del servizio tramite la console AWS

Puoi verificare che ElastiCache i tuoi cluster Redis stiano eseguendo l'ultimo aggiornamento del servizio seguendo questi passaggi:

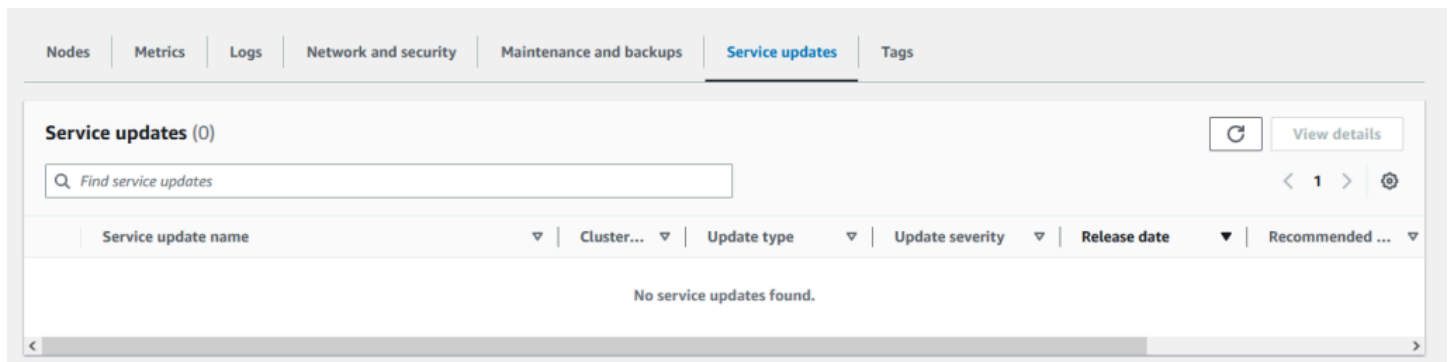
1. Scegli un cluster applicabile nella pagina Redis Clusters
2. Scegli Aggiornamenti del servizio nel riquadro di navigazione per visualizzare gli eventuali aggiornamenti del servizio applicabili per quel cluster.

Se la console visualizza un elenco di aggiornamenti del servizio, puoi selezionare l'aggiornamento del servizio e scegliere **Applica ora**.



Service update name	Cluster update status	Update type	Update severity	Release date	Recommended apply-by	Status	Cluster
elasticsearch-redis-6-2-6-update-20230109	Not-applied	engine-update	Medium	January 17, 2023, 00:00:00...	March 18, 2023, 00:59:59 (...)	Available	January 17...
elasticsearch-redis-6-2-6-patch-update	Complete	engine-update	Important	August 12, 2022, 06:00:00 ...	September 11, 2022, 05:59...	Available	December ...
elasticsearch-redis-6-2-update	Complete	engine-update	Medium	February 15, 2022, 03:00:0...	May 16, 2022, 03:59:59 (JT...	Available	March 1, 2...

Se sulla console viene visualizzato il messaggio «Nessun aggiornamento del servizio trovato», significa che al cluster ElastiCache for Redis è già applicato l'ultimo aggiornamento del servizio.



Interruzione degli aggiornamenti del servizio

Se necessario, puoi interrompere gli aggiornamenti ai cluster. Ad esempio, puoi interrompere gli aggiornamenti se si verifica un aumento imprevisto sui cluster che sono in fase di aggiornamento. Oppure potresti voler interrompere gli aggiornamenti se stanno impiegando troppo tempo e stanno interrompendo l'attività aziendale in un momento di picco.

L'operazione [Stopping \(Interruzione in corso\)](#) interrompe immediatamente tutti gli aggiornamenti a quei cluster e a tutti i nodi che devono ancora essere aggiornati. Tale operazione continua a completare tutti i nodi che hanno uno stato in progress (in corso). Tuttavia, cessa gli aggiornamenti ad altri nodi nello stesso cluster che hanno uno stato update available (aggiornamento disponibile) e li riporta a uno stato Stopping (Interruzione in corso).

Quando il flusso di lavoro Stopping (Interruzione in corso) è completo i nodi hanno uno stato che passa da Stopping (Interruzione in corso) allo stato Stopped (Interrotto). A seconda del flusso di lavoro dell'aggiornamento, alcuni cluster non avranno alcun nodo aggiornato. Altri cluster potrebbero includere alcuni nodi che vengono aggiornati e altri che presentano ancora uno stato update available (aggiornamento disponibile).

Puoi riprendere successivamente per completare il processo di aggiornamento come consentito dalle attività aziendali. In questo caso, seleziona i cluster applicabili su cui desidera completare gli aggiornamenti, quindi scegliere Apply Now (Applica ora). Per ulteriori informazioni, consulta [Applicazione degli aggiornamenti di servizio](#).

Utilizzo della console

È possibile interrompere un aggiornamento del servizio utilizzando la ElastiCache console. Di seguito viene illustrato come eseguire questa operazione:

- Dopo l'avanzamento di un aggiornamento del servizio su un cluster selezionato, la ElastiCache console visualizza la scheda Visualizza/Stop Update nella parte superiore della dashboard. ElastiCache
- Per interrompere l'aggiornamento, seleziona Stop Update (Interrompi aggiornamento).
- Quando interrompi l'aggiornamento, seleziona il cluster ed esamina lo stato. Lo stato viene modificato in Stopping (Interruzione in corso) ed eventualmente nello stato Stopped (Interrotto).

Utilizzando il AWS CLI

Puoi interrompere un aggiornamento del servizio utilizzando la AWS CLI. L'esempio di codice seguente mostra come eseguire tale operazione.

Per un gruppo di replica, procedi come descritto di seguito:

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --replication-group-ids my-replication-group-1 my-replication-group-2
```

Per un cluster di cache, procedi come descritto di seguito:

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --cache-cluster-ids my-cache-cluster-1 my-cache-cluster-2
```

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

Vulnerabilità ed esposizioni comuni (CVE): vulnerabilità di sicurezza risolte in per Redis ElastiCache

CVE (Common Vulnerabilities and Exposures - Vulnerabilità e rischi comuni) è un elenco di voci per vulnerabilità di sicurezza informatica pubblicamente note. Ogni voce è un link che contiene un numero di identificazione, una descrizione e almeno un riferimento pubblico. In questa pagina puoi trovare un elenco delle vulnerabilità di sicurezza che sono state ElastiCache risolte in Redis.

Ti consigliamo di eseguire sempre l'aggiornamento alla versione più recente di Redis ElastiCache per proteggerti dalle vulnerabilità note. Quando si utilizza una cache ElastiCache Serverless, le correzioni CVE vengono applicate automaticamente alla cache. Quando si utilizzano cluster progettati autonomamente, ElastiCache for Redis espone il componente PATCH. Ad esempio, quando si utilizza ElastiCache per la versione Redis 6.2.6, la versione principale è 6, la versione secondaria è

2 e la versione patch è 6. Le versioni PATCH riguardano correzioni di bug, correzioni di sicurezza e modifiche non funzionali compatibili con le versioni precedenti.

Puoi utilizzare questa pagina per verificare se una particolare versione di ElastiCache for Redis ha una correzione per una specifica vulnerabilità di sicurezza. Se il tuo cluster ElastiCache for Redis esegue una versione senza la correzione di sicurezza, consulta la tabella seguente e agisci. Puoi eseguire l'aggiornamento a una versione più recente ElastiCache di Redis contenente la correzione oppure, se utilizzi una versione ElastiCache per Redis contenente la correzione, assicurati di avere applicato l'ultimo aggiornamento del servizio facendo riferimento a [Gestione degli aggiornamenti del servizio](#). Per ulteriori informazioni sulle versioni supportate ElastiCache per il motore Redis e su come eseguire l'aggiornamento, consulta [Versioni e aggiornamenti del motore](#).

Note

- Se un CVE è indirizzato in una versione ElastiCache per Redis, significa che è indirizzato anche nelle versioni più recenti. Quindi, ad esempio, se una vulnerabilità viene risolta nella versione 6.0.5 ElastiCache di Redis, questa situazione continua nelle versioni 6.2.6, 7.0.7 e 7.1.
- Un asterisco (*) nella tabella seguente indica che è necessario che sia applicato l'ultimo aggiornamento del servizio per il cluster for Redis che esegue la versione ElastiCache per Redis specificata ElastiCache per risolvere la vulnerabilità di sicurezza. Per ulteriori informazioni su come verificare che sia stato applicato l'ultimo aggiornamento del servizio per la versione ElastiCache per Redis su cui è in esecuzione il cluster, consulta [Gestione degli aggiornamenti del servizio](#).

ElastiCache per la versione Redis	CVE indirizzati
Redis 6.0.5	CVE-2022-24735 *, CVE-2022-24736 *
Redis 6.2.6	CVE-2022-24834 *, CVE-2022-35977 *, CVE-2022-36021 *, CVE-2022-24735 , CVE-2022-24736
Redis 7.0.7	CVE-2023-41056 *, CVE-2022-24834 *, CVE-2022-35977 , CVE-2022-36021 , CVE-2022-24735 , CVE-2022-24736

ElastiCache per la versione Redis	CVE indirizzati
Redis 7.1.0	CVE-2023-41056 , CVE-2022-24834 , CVE-2022-35977 , CVE-2022-36021 , CVE-2022-24735 , CVE-2022-24736

Registrazione di log e monitoraggio in Amazon ElastiCache

Ai fini della gestione è importante conoscere le prestazioni delle cache. ElastiCache genera le metriche pubblicate in File di log Amazon CloudWatch per il monitoraggio delle prestazioni delle cache. Inoltre, ElastiCache genera eventi quando si verificano modifiche significative sulle risorse di cache (ad esempio, viene creata una nuova cache o viene eliminata una cache).

Argomenti

- [Metriche ed eventi serverless](#)
- [Metriche ed eventi di cluster progettati autonomamente](#)
- [Registrazione delle chiamate all'API Amazon ElastiCache con AWS CloudTrail](#)

Metriche ed eventi serverless

Questa sezione descrive le metriche e gli eventi che puoi monitorare quando usi le cache serverless.

Argomenti

- [Metriche delle cache serverless](#)
- [Eventi di cache serverless](#)

Metriche delle cache serverless

Lo spazio dei nomi AWS/ElastiCache include le seguenti metriche CloudWatch per le cache serverless Redis.

Parametro	Descrizione	Unità
BytesUsedForCache	Numero totale di byte utilizzati dai dati archiviati nella cache.	Byte
ElastiCacheProcessingUnits	Numero totale di ElastiCacheProcessingUnits (ECPU) usate dalle richieste eseguite nella cache.	Conteggio

Parametro	Descrizione	Unità
SuccessfulReadRequestLatency	Latenza delle richieste di lettura completate.	Microsecondi
SuccessfulWriteRequestLatency	Latenza delle richieste di scrittura completate.	Microsecondi
TotalCmdsCount	Numero totale dei comandi eseguiti nella cache.	Conteggio
CacheHitRate	Indica la percentuale di riscontri nella cache. Questo viene calcolato utilizzando le statistiche <code>cache_hits</code> e <code>cache_misses</code> nel modo seguente: $\text{cache_hits} / (\text{cache_hits} + \text{cache_misses})$.	Percentuale
CacheHits	Il numero di ricerche di chiavi di sola lettura nella cache.	Conteggio
CurrConnections	Numero di connessioni client alla cache.	Conteggio
ThrottledCmds	Numero di richieste sottoposte da ElastiCache alla limitazione (della larghezza di banda della rete) perché il carico di lavoro veniva dimensionato più velocemente di quanto ElastiCache potesse fare.	Conteggio
NewConnections	Il numero totale di connessioni accettate dal server durante questo periodo.	Conteggio

Parametro	Descrizione	Unità
Currltems	Il numero di elementi nella cache.	Conteggio
CurrVolatileItems	Il numero di elementi nella cache con TTL.	Conteggio
NetworkBytesIn	Byte totali trasferiti nella cache.	Byte
NetworkBytesOut	Byte totali trasferiti dalla cache.	Byte
Espulsioni	Numero di chiavi eliminate dalla cache.	Conteggio
IamAuthenticationExpirations	Il numero totale di connessioni Redis autenticate da IAM scadute. Ulteriori informazioni su Autenticazione con IAM sono disponibili nella guida per l'utente.	Conteggio
IamAuthenticationThrottling	Il numero totale di richieste Redis AUTH o HELLO autenticate da IAM con throttling. Ulteriori informazioni su Autenticazione con IAM sono disponibili nella guida per l'utente.	Conteggio

Parametro	Descrizione	Unità
KeyAuthorizationFailures	Numero totale di tentativi non riusciti da parte degli utenti di accedere a chiavi a cui non hanno l'autorizzazione ad accedere. Consigliamo di impostare un allarme per rilevare i tentativi di accesso non autorizzato.	Conteggio
AuthenticationFailures	Numero totale di tentativi non riusciti di autenticazione su Redis utilizzando il comando AUTH. Consigliamo di impostare un allarme per rilevare i tentativi di accesso non autorizzato.	Conteggio
CommandAuthorizationFailures	Numero totale di tentativi non riusciti da parte degli utenti di eseguire comandi che non dispongono dell'autorizzazione e per chiamare. Consigliamo di impostare un allarme per rilevare i tentativi di accesso non autorizzato.	Conteggio

Metriche a livello di comando

ElastiCache genera anche le seguenti metriche a livello di comando. Per ogni tipo di comando, ElastiCache restituisce il numero totale di comandi e il numero di ECPU utilizzate dal tipo di comando.

Parametro	Descrizione	Unità
EvalBasedCmds	Il numero di comandi get che la cache ha ricevuto.	Conteggio
EvalBasedCmdsECPUs	ECPU utilizzate dai comandi basati su valutazione.	Conteggio
GeoSpatialBasedCmds	Numero totale di comandi per i comandi basati su GeoSpace. È ricavato dalla statistica Redis commandstats sommando tutti i tipi di comandi geo: geoadd, geodist, geohash, geopos, georadius e georadiusbymember.	Conteggio
GeoSpatialBasedCmdsECPUs	ECPU utilizzate dai comandi basati su dati geospaziali.	Conteggio
GetTypeCmds	Il numero totale di comandi di tipo sola lettura. È ricavato dalla statistica Redis commandstats sommando tutti i comandi di tipo sola lettura (get, hget, scard, lrange e così via).	Conteggio
GetTypeCmdsECPUs	ECPU utilizzate dai comandi di lettura.	Conteggio
HashBasedCmds	Il numero totale di comandi basati su hash. È ricavato dalla statistica Redis commandstats sommando tutti i comandi che agiscono su uno o più hash (hget, hkeys, hvals, hdel e così via).	Conteggio

Parametro	Descrizione	Unità
HashBasedCmdsECPUs	ECPU utilizzate dai comandi basati su hash.	Conteggio
HyperLogLogBasedCmds	Il numero totale di comandi basati su HyperLogLog. È ricavato dalla statistica Redis <code>commandstats</code> sommando tutti i comandi di tipo <code>pf</code> (<code>pfadd</code> , <code>pfcount</code> , <code>pfmerge</code> e così via).	Conteggio
HyperLogLogBasedCmdsECPUs	ECPU utilizzate dai comandi basati su HyperLogLog.	Conteggio
JsonBasedCmds	Il numero totale di comandi JSON, inclusi i comandi di lettura e scrittura. È ricavato dalla statistica Redis <code>commandstats</code> sommando tutti i comandi JSON che agiscono sulle chiavi JSON.	Conteggio
JsonBasedCmdsECPUs	ECPU utilizzate da tutti i comandi JSON, inclusi i comandi di lettura e scrittura.	Conteggio
JsonBasedGetCmds	Il numero totale di comandi JSON di sola lettura. È ricavato dalla statistica Redis <code>commandstats</code> sommando tutti i comandi di lettura JSON che agiscono sulle chiavi JSON.	Conteggio
JsonBasedGetCmdsECPUs	ECPU utilizzate dai comandi di sola lettura JSON.	Conteggio

Parametro	Descrizione	Unità
JsonBasedSetCmds	Il numero totale di comandi di scrittura JSON. È ricavato dalla statistica Redis <code>commandstats</code> sommando tutti i comandi di scrittura JSON che agiscono sulle chiavi JSON.	Conteggio
JsonBasedSetCmdsECPUs	ECPU utilizzate dai comandi di scrittura JSON.	Conteggio
KeyBasedCmds	Il numero totale di comandi basati su chiavi. È ricavato dalla statistica Redis <code>commandstats</code> sommando tutti i comandi che agiscono su una o più chiavi in più strutture di dati (<code>del</code> , <code>expire</code> , <code>rename</code> e così via).	Conteggio
KeyBasedCmdsECPUs	ECPU utilizzate dai comandi basati su chiave.	Conteggio
ListBasedCmds	Il numero totale di comandi basati su elenchi. È ricavato dalla statistica Redis <code>commandstats</code> sommando tutti i comandi che agiscono su uno o più elenchi (<code>lindex</code> , <code>lrange</code> , <code>lpush</code> , <code>ltrim</code> e così via).	Conteggio
ListBasedCmdsECPUs	ECPU utilizzate dai comandi basati su elenco.	Conteggio

Parametro	Descrizione	Unità
NonKeyTypeCmds	Il numero totale di comandi non basati su chiavi. È ricavato dalla statistica Redis <code>commandstats</code> sommando tutti i comandi che non agiscono su una o più chiavi, ad esempio <code>acl</code> , <code>dbsize</code> o <code>info</code> .	Conteggio
NonKeyTypeCmdsECPUs	ECPU utilizzate da comandi non basati su chiave.	Conteggio
PubSubBasedCmds	Numero totale di comandi per la funzionalità pub/sub. È ricavato dalla statistica Redis <code>commandstats</code> sommando tutti i comandi utilizzati per la funzionalità pub/sub: <code>punsubscribe</code> , <code>publish</code> , <code>pubsub</code> , <code>punsubscribe</code> , <code>ssubscribe</code> e, <code>unsubscribe</code> , <code>spublish</code> , <code>subscribe</code> e <code>unsubscribe</code> .	Conteggio
PubSubBasedCmdsECPUs	ECPU utilizzate dai comandi basati su pub/sub.	Conteggio
SetBasedCmds	Il numero totale di comandi basati su set. È ricavato dalla statistica Redis <code>commandstats</code> sommando tutti i comandi che agiscono su uno o più set (<code>scard</code> , <code>sdiff</code> , <code>sadd</code> , <code>sunion</code> e così via).	Conteggio
SetBasedCmdsECPUs	ECPU utilizzate dai comandi basati su set.	Conteggio

Parametro	Descrizione	Unità
SetTypeCmds	Il numero totale di comandi di tipo scrittura. È ricavato dalla statistica Redis <code>commandstats</code> sommando tutti i comandi di tipo mutativo che operano sui dati (<code>set</code> , <code>hset</code> , <code>sadd</code> , <code>lpop</code> e così via).	Conteggio
SetTypeCmdsECPUs	ECPU utilizzate dai comandi di scrittura.	Conteggio
SortedSetBasedCmds	Il numero totale di comandi basati su set ordinati. È ricavato dalla statistica Redis <code>commandstats</code> sommando tutti i comandi che agiscono su uno o più set ordinati (<code>zcount</code> , <code>zrange</code> , <code>zrank</code> , <code>zadd</code> e così via).	Conteggio
SortedSetBasedCmdsECPUs	ECPU utilizzate dai comandi basati su ordinamento.	Conteggio
StringBasedCmds	Il numero totale di comandi basati su stringhe. È ricavato dalla statistica Redis <code>commandstats</code> sommando tutti i comandi che agiscono su una o più stringhe (<code>strlen</code> , <code>setex</code> , <code>setrange</code> e così via).	Conteggio
StringBasedCmdsECPUs	ECPU utilizzate dai comandi basati su stringa.	Conteggio

Parametro	Descrizione	Unità
StreamBasedCmds	Il numero totale di comandi basati sul flusso. È ricavato dalla statistica Redis <code>commandstats</code> sommando tutti i comandi che agiscono su una o più tipi di dati di flusso (<code>xrange</code> , <code>xlen</code> , <code>xadd</code> , <code>xdel</code> e così via).	Conteggio
StreamBasedCmdsECPUs	ECPU utilizzate dai comandi basati su flusso.	Conteggio

Eventi di cache serverless

ElastiCache registra i log eventi correlati alla cache serverless. Queste informazioni includono la data, l'ora, il nome e tipo di fonte e una descrizione dell'evento. Gli eventi possono essere facilmente recuperati dal log utilizzando la console ElastiCache, il comando `describe-events` della AWS CLI o l'operazione `DescribeEvents` dell'API ElastiCache.

Puoi scegliere di monitorare, inserire, trasformare e agire sugli eventi ElastiCache utilizzando Amazon EventBridge. Ulteriori informazioni sono disponibili nella Guida per l'utente di Amazon EventBridge all'indirizzo <https://docs.aws.amazon.com/eventbridge/latest/userguide/>.

Visualizzazione degli eventi ElastiCache (console)

Per visualizzare gli eventi utilizzando la console ElastiCache:

1. Accedi alla AWS Management Console e apri la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Per l'elenco di tutti gli eventi disponibili, scegliere **Events (Eventi)** nel pannello di navigazione.
3. Nella schermata **Eventi**, ogni riga dell'elenco rappresenta un singolo evento e indica l'origine, il tipo, l'ora GMT 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 ElastiCache (AWS CLI)

Per generare un elenco di eventi ElastiCache con la AWS CLI, utilizza il 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 di cache serverless.

```
aws elasticache describe-events --source-type serverless-cache --max-items 40
```

Il codice seguente elenca tutti gli eventi di cache serverless delle ultime 24 ore (1440 minuti).

```
aws elasticache describe-events --source-type serverless-cache --duration 1440
```

Eventi serverless

Questa sezione documenta i diversi tipi di eventi che potresti ricevere per le cache serverless.

Eventi di creazione di cache serverless

Tipo di evento	Descrizione	Unità	Origine	Messaggio
Creazione della cache completata	ARN della cache	creazione	Cache serverless	La cache <nome-cache> è stata creata ed è pronta per l'uso.
Creazione della cache completata	ARN della cache ARN dello snapshot	creazione	Cache serverless	La cache <nome-cache> è stata creata e i dati sono stati ripristinati dallo snapshot. La cache è pronta per l'uso.
Creazione della cache non riuscita	ARN della cache	errore	Cache serverless	Impossibile creare la cache <nome-cache>. Gli indirizzi IP liberi sono

Tipo di evento	Descrizione	Unità	Origine	Messaggio
				insufficienti per creare un endpoint VPC.
Creazione della cache non riuscita	ARN della cache	errore	Cache serverless	Impossibile creare la cache <nome-cache>. Nella richiesta sono state fornite sottoreti non valide.
Creazione della cache non riuscita	ARN della cache	errore	Cache serverless	Impossibile creare la cache <nome-cache>. Il limite di quota è stato raggiunto per la creazione di un endpoint VPC.
Creazione della cache non riuscita	ARN della cache	errore	Cache serverless	Impossibile creare la cache <nome-cache>. Non si dispone delle autorizzazioni per creare un endpoint VPC.

Tipo di evento	Descrizione	Unità	Origine	Messaggio
Creazione della cache non riuscita	ARN della cache	errore	Cache serverless	Impossibile creare la cache <nome-cache>. Un utente con una versione Redis incompatibile è presente nel gruppo di utenti <nome-gruppo-utenti>.
Creazione della cache non riuscita	ARN della cache ARN dello snapshot della cache	errore	Cache serverless	Impossibile creare la cache <nome-cache>. Il gruppo di utenti fornito <nome-gruppo-utenti> non esiste.

Tipo di evento	Descrizione	Unità	Origine	Messaggio
Creazione della cache non riuscita	ARN della cache	errore	Cache serverless	<p>Impossibile creare la cache <nome-cache>. Il ripristino dei dati dallo snapshot non è riuscito perché <motivo>.</p> <p>Motivi dell'errore:</p> <ul style="list-style-type: none"> recupero del file da S3 non riuscito; l'hash md5 previsto non corrisponde a quello effettivo ; il file RDB fornito ha una versione non supportata.

Eventi di aggiornamento di cache serverless

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
Aggiornamento della cache completato	ARN della cache	modifica della configurazione	Cache serverless	SecurityGroups aggiornati per la cache <nome-cache>.

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
Aggiornamento della cache completato	ARN della cache	modifica della configurazione	Cache serverless	Tag aggiornati per la cache <nome-cache>.
Aggiornamento della cache non riuscito	ARN della cache	modifica della configurazione	Cache serverless	L'aggiornamento della cache <nome-cache> non è riuscito. Un utente con una versione Redis incompatibile è presente nel gruppo di utenti <nome-gruppo-utenti>.
Aggiornamento della cache non riuscito	ARN della cache	modifica della configurazione	Cache serverless	L'aggiornamento della cache <nome-cache> non è riuscito. L'aggiornamento di SecurityGroups non è riuscito.

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
Aggiornamento della cache non riuscito	ARN della cache	modifica della configurazione	Cache serverless	L'aggiornamento della cache <nome-cache> non è riuscito. L'aggiornamento di SecurityGroups non è riuscito a causa di autorizzazioni insufficienti.
Aggiornamento della cache non riuscito	ARN della cache	modifica della configurazione	Cache serverless	L'aggiornamento della cache <nome-cache> non è riuscito. L'aggiornamento di SecurityGroups non è riuscito perché i SecurityGroups non sono validi.

Eventi di eliminazione di cache serverless

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
Eliminazione della cache completata	ARN della cache	eliminazione	Cache serverless	La cache <nome-cache> è stata eliminata.

Eventi di limite di utilizzo di cache serverless

Tipo di evento	Descrizione	Unità	Origine	Messaggio
Aggiornamento della cache completato	ARN della cache	modifica della configurazione	Cache serverless	Limiti aggiornati per la cache <nome-cache>.
Avvicinamento al limite della cache	ARN della cache	notification	Cache serverless	Lo slot <X> utilizza più del <Y>% del limite per slot di 32 GB. Ad esempio, Lo slot 10 utilizza più del 90% del limite per slot di 32 GB.
Aggiornamento della cache non riuscito	ARN della cache	errore	Cache serverless	L'aggiornamento di un limite per la cache <nome-cache> non è riuscito perché la cache è stata eliminata.
Aggiornamento della cache non riuscito	ARN della cache	errore	Cache serverless	L'aggiornamento di un limite per la cache <nome-cache> non è riuscito a causa di una configurazione non valida.
Aggiornamento della cache non riuscito	ARN della cache	errore	Cache serverless	Un aggiornamento del limite per la cache <nome-cache> non è

Tipo di evento	Descrizione	Unità	Origine	Messaggio
				riuscito perché i dati correnti memorizzati nella cache superano i nuovi limiti. Pulisci i dati prima di applicare i limiti.

Eventi di snapshot della cache serverless

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
Creazione dello snapshot completata	ARN della cache ARN dello snapshot	creazione	serverless-cache-snapshot	Lo snapshot <nome-snapshot> è stato creato per la cache <nome-cache>.
Creazione dello snapshot non riuscita	ARN della cache ARN dello snapshot	errore	serverless-cache-snapshot	Impossibile creare lo snapshot per la cache <nome-cache>. Creazione dello snapshot <nome-snapshot> non riuscita con la chiave gestita dal cliente <id-chiave> <motivo>.

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
				<p>Messaggi del motivo dell'errore:</p> <ul style="list-style-type: none"> • perché la chiave gestita dal cliente è disattivata; • perché la chiave gestita dal cliente non è stata trovata; • perché si è verificato il timeout della richiesta.
Creazione dello snapshot non riuscita	<p>ARN della cache</p> <p>ARN dello snapshot</p>	errore	serverless-cache-snapshot	<p>Impossibile creare lo snapshot per la cache <nome-cache>. La creazione dello snapshot <nome-snapshot> non è riuscita perché <motivo>.</p> <p>Motivo predefinito:</p> <ul style="list-style-type: none"> • a causa di un errore interno.

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
Esportazione dello snapshot non riuscita	ARN dello snapshot	errore	serverless-cache-snapshot	Impossibile esportare lo snapshot per la cache <nome-cache>. Impossibile esportare lo snapshot nel bucket %s perché ElastiCache non dispone delle autorizzazioni per il bucket.
Esportazione dello snapshot non riuscita	ARN dello snapshot	errore	serverless-cache-snapshot	Impossibile esportare lo snapshot per la cache <nome-cache>. Impossibile esportare lo snapshot nel bucket "%s" perché nel bucket è già presente un oggetto con lo stesso nome.

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
Esportazione dello snapshot non riuscita	ARN dello snapshot	errore	serverless-cache-snapshot	Impossibile esportare lo snapshot per la cache <nome-cache>. Impossibile esportare lo snapshot nel bucket "%s" perché l'ID dell'account del proprietario del bucket è cambiato.
Esportazione dello snapshot non riuscita	ARN dello snapshot	errore	serverless-cache-snapshot	Impossibile esportare lo snapshot per la cache <nome-cache>. Impossibile esportare lo snapshot nel bucket "%s" perché il bucket S3 non è accessibile.

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
Esportazione dello snapshot non riuscita	ARN dello snapshot	errore	serverless-cache-snapshot	Impossibile esportare lo snapshot per la cache <nome-cache>. Impossibile esportare lo snapshot nel bucket "%s" perché il bucket non è accessibile.
Esportazione dello snapshot non riuscita	ARN dello snapshot	errore	serverless-cache-snapshot	Impossibile esportare lo snapshot per la cache <nome-cache>. Impossibile esportare lo snapshot nel bucket "%s" perché il bucket non esiste.

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
Esportazione dello snapshot non riuscita	ARN dello snapshot	errore	serverless-cache-snapshot	Impossibile esportare lo snapshot per la cache <nome-cache>. Impossibile esportare lo snapshot nel bucket "%s" con la chiave gestita dal cliente dello snapshot di origine %s <motivo>.
Esportazione dello snapshot non riuscita	ARN dello snapshot	errore	serverless-cache-snapshot	Impossibile esportare lo snapshot per la cache <nome-cache>. Impossibile esportare lo snapshot nel bucket "%s".

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
Copia dello snapshot non riuscita	ARN dello snapshot 1 ARN dello snapshot 2	errore	serverless-cache-snapshot	Impossibile copiare lo snapshot <nome-snapshot>. Impossibile copiare lo snapshot "%s" nello snapshot "%s" con la chiave gestita dal cliente dello snapshot di origine <id-chiave> <nome-motivo>.
Copia dello snapshot non riuscita	ARN dello snapshot 1 ARN dello snapshot 2	errore	serverless-cache-snapshot	Impossibile copiare lo snapshot <nome-snapshot>. Impossibile copiare lo snapshot "%s" nello snapshot "%s" con la chiave gestita dal cliente dello snapshot di destinazione "%s" "%s".

Metriche ed eventi di cluster progettati autonomamente

In questa sezione sono descritti le metriche, gli eventi e i log previsti quando si usano i cluster progettati autonomamente.

Argomenti

- [Metriche per cluster progettati autonomamente](#)
- [Eventi per cluster progettati autonomamente](#)
- [Consegna dei registri](#)
- [Monitoraggio dell'utilizzo con i parametri di CloudWatch](#)
- [Monitoraggio di Amazon SNS per gli eventi ElastiCache](#)

Metriche per cluster progettati autonomamente

Quando si progettano autonomamente i cluster, ElastiCache genera le metriche a ogni livello di nodo, incluse quelle a livello di host e della cache.

Per ulteriori informazioni sulle metriche a livello di host, consulta [Parametri a livello di host](#).

Per ulteriori informazioni sulle metriche a livello di nodo, consulta [Parametri di Redis](#).

Eventi per cluster progettati autonomamente

ElastiCache registra i log eventi correlati alle cache progettate autonomamente. Quando usi i cluster progettati autonomamente, puoi visualizzare gli eventi del cluster nella console ElastiCache utilizzando la AWS CLI o Amazon Simple Notification Service (SNS). Gli eventi dei cluster progettati autonomamente non vengono pubblicati su Amazon EventBridge.

Le informazioni degli eventi dei cluster progettati autonomamente includono la data, l'ora, il nome e tipo di origine e una descrizione dell'evento. Gli eventi possono essere facilmente recuperati dal log utilizzando la console ElastiCache, il comando `describe-events` della AWS CLI o l'operazione `DescribeEvents` dell'API ElastiCache.

Visualizzazione degli eventi ElastiCache (Console)

La seguente procedura consente di visualizzare eventi con la console ElastiCache.

Per visualizzare eventi utilizzando la console ElastiCache

1. Accedi alla AWS Management Console e apri la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Per l'elenco di tutti gli eventi disponibili, scegliere Events (Eventi) nel pannello di navigazione.
3. Nella schermata Eventi, ogni riga dell'elenco rappresenta un singolo evento e indica l'origine, il tipo, l'ora GMT 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 ElastiCache (AWS CLI)

Per generare un elenco di eventi ElastiCache con la AWS CLI, utilizza il 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 di cluster progettato autonomamente.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

Il codice seguente elenca tutti gli eventi di cache progettate autonomamente delle ultime 24 ore (1440 minuti).

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

Eventi di cluster progettati autonomamente


Questa sezione contiene l'elenco degli eventi che puoi ricevere per i cluster progettati autonomamente.


I seguenti eventi ElastiCache attivano notifiche Amazon SNS. Per ulteriori informazioni sui dettagli degli eventi, consulta [Visualizzazione di eventi ElastiCache C](#).

Nome evento	Messaggio	Descrizione
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	Un nodo di cache è stato aggiunto al cluster di cache ed è pronto per l'uso.

Nome evento	Messaggio	Descrizione
ElastiCache:AddCacheNodeFailed due to insufficient free IP addresses	ElastiCache:AddCacheNodeFailed : <i>cluster-name</i>	Non è stato possibile aggiungere un nodo di cache per l'insufficienza di indirizzi IP disponibili.
ElastiCache:CacheClusterParametersChanged	ElastiCache:CacheClusterParametersChanged : <i>cluster-name</i>	Uno o più parametri del cluster di cache sono stati modificati.
Elasticache:cacheclusterProvisioningComplete	ElastiCache:CacheClusterProvisioningComplete <i>cluster-name-0001-005</i>	Il provisioning di un cluster di cache è completo e i nodi di cache nel cluster sono pronti all'uso.
ElastiCache:CacheClusterProvisioningFailed due to incompatible network state	ElastiCache:CacheClusterProvisioningFailed : <i>cluster-name</i>	È stato effettuato un tentativo di avvio di un nuovo cluster di cache in un cloud privato virtuale (VPC, Virtual Private Cloud) inesistente.
ElastiCache:CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	Dimensionamento di un cluster di cache completato.
Elasticache: cacheclusterScalingFailed	ElasticaCacheclusterScalingFailed: <i>nome-cluster</i>	Non è riuscita l'operazione di dimensionamento di un cluster di cache.

Nome evento	Messaggio	Descrizione
Elasticache:cacheclusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	<p>Si è verificato uno dei seguenti eventi:</p> <ul style="list-style-type: none">• L'elenco dei gruppi di sicurezza cache autorizzati per il cluster di cache è stato modificato.• Uno o più nuovi gruppi di sicurezza EC2 sono stati ammessi nei gruppi di sicurezza cache associati al cluster di cache.• Uno o più gruppi di sicurezza EC2 sono stati rimossi dai gruppi di sicurezza cache associati al cluster di cache.

Nome evento	Messaggio	Descrizione
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	<p>ElastiCache ha rilevato che l'host che esegue un nodo di cache è danneggiato o irraggiungibile ed è iniziata la sostituzione di tale nodo.</p> <div data-bbox="1068 495 1507 806"><p> Note</p><p>La voce DNS relativa al nodo di cache in sostituzione non cambia.</p></div> <p>Nella maggior parte dei casi, non è necessario aggiornare l'elenco dei server per i client, quando si verifica questo evento. Tuttavia, alcune librerie client della cache potrebbero smettere di utilizzare il nodo di cache nonostante la sostituzione da parte di ElastiCache; in tal caso, l'applicazione dovrà aggiornare l'elenco dei server.</p>

Nome evento	Messaggio	Descrizione
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	<p>ElastiCache ha rilevato che l'host che esegue un nodo di cache è danneggiato o irraggiungibile ed effettuato la sostituzione di tale nodo.</p> <div data-bbox="1068 495 1507 806" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note</p> <p>La voce DNS relativa al nodo di cache in sostituzione non cambia.</p> </div> <p>Nella maggior parte dei casi, non è necessario aggiornare l'elenco dei server per i client, quando si verifica questo evento. Tuttavia, alcune librerie client della cache potrebbero smettere di utilizzare il nodo di cache nonostante la sostituzione da parte di ElastiCache; in tal caso, l'applicazione dovrà aggiornare l'elenco dei server.</p>
ElastiCache:cacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	<p>Uno o più nodi di cache sono stati riavviati.</p> <p>Messaggio (Memcached): "Cache node %s shutdown" Secondo messaggio: "Cache node %s restarted"</p>

Nome evento	Messaggio	Descrizione
ElastiCache: CertificateRenewalComplete (solo Redis)	ElastiCache:CertificateRenewalComplete	Il certificato Amazon CA è stato rinnovato correttamente.
ElastiCache:CreateReplicationGroupComplete	ElastiCache:CreateReplicationGroupComplete : <i>cluster-name</i>	Il gruppo di replica è stato creato correttamente.
ElastiCache>DeleteCacheClusterComplete	ElastiCache>DeleteCacheClusterComplete : <i>cluster-name</i>	È stata completata la procedura di eliminazione di un cluster di cache con tutti i nodi associati.
ElastiCache:FailoverComplete (solo Redis)	ElastiCache:FailoverComplete : <i>mycluster</i>	Il failover su un nodo di replica ha avuto esito positivo.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	Il numero di repliche nel cluster è stato aumentato.
Elasticache:ReplicationGroupIncreaseReplicaCount	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	Il processo di aggiunta di repliche al cluster è iniziato.
Elasticache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	È stata annullata la sostituzione programmata di un nodo nel cluster.

Nome evento	Messaggio	Descrizione
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	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 .
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	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 .
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Un nodo di cache è stato rimosso dal cluster di cache.
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	Operazione di scalabilità verticale sul gruppo di replica completata correttamente.
ElastiCache:ReplicationGroupScalingFailed	"Failed applying modification to cache node type to %s."	Operazione di scalabilità verticale sul gruppo di replica non riuscita.
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	È disponibile un aggiornamento self-service per il nodo.

Nome evento	Messaggio	Descrizione
ElastiCache: SnapshotComplete (solo Redis)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	È stata correttamente acquisita una snapshot di cache.
ElastiCache: SnapshotFailed (solo Redis)	SnapshotFailed : <i>cluster-name</i>	<p>Non è stata portata a compimento l'acquisizione di una snapshot di cache. Per informazioni dettagliate sulle cause, consulta gli eventi della cache del cluster.</p> <p>Lo stato della snapshot, riportato in DescribeSnapshots, sarà failed.</p>

Consegna dei registri

Note

Il registro lento di Redis è supportato per gruppi di replica e cluster di cache Redis che utilizzano la versione 6.0 e successive del motore.

Il registro motore di Redis è supportata per gruppi di replica e cluster di cache Redis che utilizzano la versione 6.2 e successive del motore.

La consegna dei registri consente il flusso Redis [SLOWLOG](#) o Registro motore Redis in una di due destinazioni:

- Amazon Data Firehose
- CloudWatch Registri Amazon

Puoi abilitare e configurare la consegna dei log quando crei o modifichi un cluster utilizzando le ElastiCache API. Ogni voce del registro verrà recapitata alla destinazione specificata in uno dei due formati: JSON o TEXT.

Un numero fisso di voci di registro lento viene recuperato periodicamente dal motore Redis. A seconda del valore specificato per il parametro del motore `slowlog-max-len`, ulteriori voci di registro lente potrebbero non essere recapitate alla destinazione.

Puoi scegliere di modificare le configurazioni di consegna o disabilitare la consegna dei log in qualsiasi momento utilizzando la AWS console o una delle API di modifica, oppure. [modify-cache-clustermodify-replication-group](#)

È necessario impostare il parametro `apply-immediately` per tutte le modifiche al recapito dei registri.

Note

CloudWatch I costi di Amazon Logs si applicano quando la consegna dei log è abilitata, anche quando i log vengono consegnati direttamente ad Amazon Data Firehose. Per ulteriori informazioni, consulta la sezione Vend Logs in [Amazon CloudWatch Pricing](#).

Contenuto di una voce di registro lenta

Il ElastiCache for Redis Slow Log contiene le seguenti informazioni:

- `CacheClusterId`— L'ID del cluster di cache
- `CacheNodeId`— L'ID del nodo di cache
- `Id` – Un identificativo progressivo univoco per ogni voce di registro lenta
- `Timestamp` – Il timestamp Unix nella quale il comando registrato è stato elaborato
- `Durata` – La quantità di tempo necessario per la sua esecuzione, in microsecondi
- `Comando` – Il comando utilizzato dal client. Ad esempio, `set foo bar foo` dov'è la chiave e dove `bar` è il valore. ElastiCache for Redis sostituisce il nome e il valore effettivi della chiave con `(2 more arguments)` per evitare di esporre dati sensibili.
- `ClientAddress`— Indirizzo IP e porta del client
- `ClientName`— Nome del client se impostato tramite il `CLIENT SETNAME` comando

Contenuto di una voce del registro motore

Il registro di ElastiCache for Redis Engine contiene le seguenti informazioni:

- `CacheClusterId`— L'ID del cluster di cache
- `CacheNodeId`— L'ID del nodo di cache
- `Livello di registro`: `LogLevel` può essere uno dei seguenti: `VERBOSE("-")`, `NOTICE("*")`, `WARNING("#")`.
- `Ora`: l'ora UTC del messaggio registrato. L'ora è nel formato seguente: `"DD MMM YYYY hh:mm:ss.ms UTC"`
- `Ruolo`: il ruolo del nodo da cui viene emesso il registro. Può essere uno dei seguenti: «M» per primario, «S» per replica, «C» per processo secondario di scrittura che lavora su RDB/AOF o «X» per sentinella.
- `Messaggio`: il messaggio del registro motore Redis.

Autorizzazioni per configurare la registrazione

È necessario includere le seguenti autorizzazioni IAM nella policy relativa all'utente/ruolo IAM:

- `logs:CreateLogDelivery`

- logs:UpdateLogDelivery
- logs>DeleteLogDelivery
- logs:GetLogDelivery
- logs>ListLogDeliveries

Per ulteriori informazioni, consulta [Panoramica delle autorizzazioni e delle policy per la gestione degli accessi](#).

Specificazioni del tipo di registro e del formato

Registro lento

Il registro lento supporta sia JSON che TEXT

Di seguito è riportato un esempio in formato JSON:

```
{
  "CacheClusterId": "logslowxxxxmsxj",
  "CacheNodeId": "0001",
  "Id": 296,
  "Timestamp": 1605631822,
  "Duration (us)": 0,
  "Command": "GET ... (1 more arguments)",
  "ClientAddress": "192.168.12.104:55452",
  "ClientName": "logslowxxxxmsxj##"
}
```

Di seguito è riportato un esempio in formato TEXT:

```
logslowxxxxmsxj,0001,1605631822,30,GET ... (1 more
arguments),192.168.12.104:55452,logslowxxxxmsxj##
```

Registro motore

Il registro lento supporta sia JSON che TEXT

Di seguito è riportato un esempio in formato JSON:

```
{
```

```
"CacheClusterId": "xxxxxxxxxzy-engine-log-test",
"CacheNodeId": "0001",
"LogLevel": "VERBOSE",
"Role": "M",
"Time": "12 Nov 2020 01:28:57.994 UTC",
"Message": "Replica is waiting for next BGSAVE before synchronizing with the primary.
Check back later"
}
```

Di seguito è riportato un esempio in formato TEXT:

```
xxxxxxxxxzy-engine-log-test/0001:M 29 Oct 2020 20:12:20.499 UTC * A slow-running Lua
script detected that is still in execution after 10000 milliseconds.
```

ElastiCache destinazioni di registrazione

Questa sezione descrive le destinazioni di registrazione che è possibile scegliere per i ElastiCache registri. Ogni sezione fornisce indicazioni per configurare la registrazione per il tipo di destinazione e informazioni su qualsiasi comportamento specifico del tipo di destinazione. Dopo aver configurato la destinazione di registrazione, puoi fornire le relative specifiche alla configurazione di ElastiCache registrazione per iniziare a registrarla.

Argomenti

- [CloudWatch Registri Amazon](#)
- [Amazon Data Firehose](#)

CloudWatch Registri Amazon

- CloudWatch Specifica un gruppo di log Logs in cui verranno consegnati i log.
- I registri di multipli cluster di Redis e i gruppi di replica possono essere recapitati allo stesso gruppo di registro.
- Verrà creato un nuovo flusso di registrazione per ogni nodo all'interno di un cluster di cache o di un gruppo di replica e i registri verranno recapitati ai rispettivi flussi di registrazione. Il nome del flusso di registrazione utilizzerà il formato seguente: `elasticache/${engine-name}/${cache-cluster-id}/${cache-node-id}/${log-type}`

Autorizzazioni per pubblicare i registri nei registri CloudWatch

È necessario disporre delle seguenti impostazioni di autorizzazione ElastiCache per configurare Redis per inviare i log a un gruppo di log Logs: CloudWatch

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs>DeleteLogDelivery",
        "logs:ListLogDeliveries"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow",
      "Sid": "ElastiCacheLogging"
    },
    {
      "Sid": "ElastiCacheLoggingCWL",
      "Action": [
        "logs:PutResourcePolicy",
        "logs:DescribeResourcePolicies",
        "logs:DescribeLogGroups"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Logs sent to Logs](#). CloudWatch

Amazon Data Firehose

- Si specifica un flusso di distribuzione Firehose in cui verranno consegnati i log.
- I registri di più cluster di Redis e gruppi di replica possono essere consegnati allo stesso flusso di consegna.

- I registri di ogni nodo all'interno di un cluster di cache o di un gruppo di replica verranno consegnati allo stesso flusso di consegna. È possibile distinguere i messaggi di registro da nodi di cache differenti in base al `cache-cluster-id` e `cache-node-id` inclusi in ogni messaggio di registro.
- La consegna dei registri a Firehose non è attualmente disponibile nella regione Asia Pacifico (Osaka).

Autorizzazioni per la pubblicazione dei log su Firehose

È necessario disporre delle seguenti autorizzazioni ElastiCache per configurare Redis per inviare i log a un flusso di distribuzione di Amazon Kinesis Data Firehose.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs>DeleteLogDelivery",
        "logs:ListLogDeliveries"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow",
      "Sid": "ElastiCacheLogging"
    },
    {
      "Sid": "ElastiCacheLoggingFHSLR",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Sid": "ElastiCacheLoggingFH",
      "Action": [
        "firehose:TagDeliveryStream"
      ],
    }
  ]
}
```

```
        "Resource": "Amazon Kinesis Data Firehose delivery stream ARN",
        "Effect": "Allow"
    }
]
}
```

Specifica della consegna del registro tramite la console

Utilizzando la AWS Management Console è possibile creare un cluster di Redis (modalità cluster disabilitata) seguendo i passaggi indicati in [Creazione di un cluster Redis \(modalità cluster disabilitata\) \(Console\)](#) o creare un cluster di Redis (modalità cluster abilitata) utilizzando i passaggi indicati in [Creazione di un cluster Redis \(modalità cluster abilitata\) \(console\)](#). In entrambi i casi, è possibile configurare la consegna dei registri eseguendo le operazioni seguenti;

1. Sotto Impostazioni avanzate di Redis, scegli Registri e seleziona Registri lenti o Registri motore.
2. In Formato del registro, scegliere Text o JSON.
3. InType di destinazione, scegliere CloudWatch Logs o Kinesis Firehose.
4. In Destinazione registro, scegli Crea nuovo e inserisci il nome del bucket Amazon S3, il nome del gruppo di log CloudWatchLogs o il nome del flusso di Kinesis Data Firehose, oppure scegli Seleziona esistente e seleziona il nome del gruppo di log CloudWatch Logs o il nome del flusso Kinesis Data Firehose,

Quando si modifica un cluster:

È possibile scegliere di abilitare/disabilitare la consegna dei registri o modificare il tipo di destinazione, il formato o la destinazione:

1. Accedere alla Console e aprire la console di ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dal riquadro di navigazione scegli Cluster Redis.
3. Nell'elenco di cluster, scegliere quello da dimensionare. Seleziona Cluster name (Nome del cluster) e non la casella di controllo accanto ad essa.
4. Sul Cluster name (Nome del cluster) Scegliere la scheda registro.
5. Per abilitare/disabilitare i registri lenti, scegliere Abilitare registri lenti o Disabilitare registri lenti.
6. Per abilitare/disabilitare i registri del motore, scegli tra Enable engine logs (Abilita registri motore) o Disable engine logs (Disabilita registri motore).

7. Per modificare la configurazione, scegli Modifica registri lenti o Modifica registri motore:

- In Tipo di destinazione, scegliere CloudWatch Log o Kinesis Firehose.
- In Destinazione registro, scegliere Crea nuovi e inserire il nome del gruppo di registro CloudWatchLogs o il nome del flusso Kinesis Data Firehose. Oppure scegliere Seleziona esistente e scegliere il nome del gruppo di registro CloudWatchLogs o il nome del flusso Kinesis Data Firehose.

Specificare la consegna dei registri utilizzando il AWS CLI

Registro lento

Creare un gruppo di replica con consegna lenta dei log ai CloudWatch registri.

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id test-slow-log \  
  --replication-group-description test-slow-log \  
  --engine redis \  
  --cache-node-type cache.r5.large \  
  --num-cache-clusters 2 \  
  --log-delivery-configurations '{  
    "LogType":"slow-log",  
    "DestinationType":"cloudwatch-logs",  
    "DestinationDetails":{  
      "CloudWatchLogsDetails":{  
        "LogGroup":"my-log-group"  
      }  
    },  
    "LogFormat":"json"  
  }'
```

Per Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id test-slow-log ^  
  --replication-group-description test-slow-log ^  
  --engine redis ^  
  --cache-node-type cache.r5.large ^  
  --num-cache-clusters 2 ^
```

```
--log-delivery-configurations '{
  "LogType":"slow-log",
  "DestinationType":"cloudwatch-logs",
  "DestinationDetails":{
    "CloudWatchLogsDetails":{
      "LogGroup":"my-log-group"
    }
  },
  "LogFormat":"json"
}'
```

Modifica un gruppo di replica per fornire log lenti a Logs CloudWatch

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
  --replication-group-id test-slow-log \
  --apply-immediately \
  --log-delivery-configurations '
{
  "LogType":"slow-log",
  "DestinationType":"cloudwatch-logs",
  "DestinationDetails":{
    "CloudWatchLogsDetails":{

      "LogGroup":"my-log-group"
    }
  },
  "LogFormat":"json"
}'
```

Per Windows:

```
aws elasticache modify-replication-group ^
  --replication-group-id test-slow-log ^
  --apply-immediately ^
  --log-delivery-configurations '
{
  "LogType":"slow-log",
  "DestinationType":"cloudwatch-logs",
  "DestinationDetails":{
    "CloudWatchLogsDetails":{
      "LogGroup":"my-log-group"
    }
  }
}'
```

```
    }  
  },  
  "LogFormat":"json"  
}'
```

Modificare un gruppo di replica per disabilitare la consegna lenta dei registri

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id test-slow-log \  
  --apply-immediately \  
  --log-delivery-configurations '  
{  
  "LogType":"slow-log",  
  "Enabled":false  
}'
```

Per Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id test-slow-log ^  
  --apply-immediately ^  
  --log-delivery-configurations '  
{  
  "LogType":"slow-log",  
  "Enabled":false  
}'
```

Registro motore

Crea un gruppo di replica con consegna dei log del motore a Logs. CloudWatch

Per Linux, macOS o Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id test-slow-log \  
  --replication-group-description test-slow-log \  
  --engine redis \  
  --cache-node-type cache.r5.large \  
  --num-cache-clusters 2 \  
  --log-delivery-configurations '{
```

```

    "LogType":"engine-log",
    "DestinationType":"cloudwatch-logs",
    "DestinationDetails":{
      "CloudWatchLogsDetails":{
        "LogGroup":"my-log-group"
      }
    },
    "LogFormat":"json"
  }'

```

Per Windows:

```

aws elasticache create-replication-group ^
  --replication-group-id test-slow-log ^
  --replication-group-description test-slow-log ^
  --engine redis ^
  --cache-node-type cache.r5.large ^
  --num-cache-clusters 2 ^
  --log-delivery-configurations '{
    "LogType":"engine-log",
    "DestinationType":"cloudwatch-logs",
    "DestinationDetails":{
      "CloudWatchLogsDetails":{
        "LogGroup":"my-log-group"
      }
    },
    "LogFormat":"json"
  }'

```

Modificare un gruppo di replica per inviare il registro del motore a Firehose

Per Linux, macOS o Unix:

```

aws elasticache modify-replication-group \
  --replication-group-id test-slow-log \
  --apply-immediately \
  --log-delivery-configurations '
{
  "LogType":"engine-log",
  "DestinationType":"kinesis-firehose",
  "DestinationDetails":{
    "KinesisFirehoseDetails":{
      "DeliveryStream":"test"
    }
  }
}'

```

```
    }  
  },  
  "LogFormat":"json"  
}'
```

Per Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id test-slow-log ^  
  --apply-immediately ^  
  --log-delivery-configurations '  
  {  
    "LogType":"engine-log",  
    "DestinationType":"kinesis-firehose",  
    "DestinationDetails":{  
      "KinesisFirehoseDetails":{  
        "DeliveryStream":"test"  
      }  
    },  
    "LogFormat":"json"  
  }'  
'
```

Modifica di un gruppo di replica per passare al formato motore

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \  
  --replication-group-id test-slow-log \  
  --apply-immediately \  
  --log-delivery-configurations '  
  {  
    "LogType":"engine-log",  
    "LogFormat":"json"  
  }'  
'
```

Per Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id test-slow-log ^  
  --apply-immediately ^  
  --log-delivery-configurations '  
  {
```

```
"LogType":"engine-log",
"LogFormat":"json"
}'
```

Modifica di un gruppo di replica per disabilitare la consegna registri motore

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
  --replication-group-id test-slow-log \
  --apply-immediately \
  --log-delivery-configurations '
{
  "LogType":"engine-log",
  "Enabled":false
}'
```

Per Windows:

```
aws elasticache modify-replication-group ^
  --replication-group-id test-slow-log ^
  --apply-immediately ^
  --log-delivery-configurations '
{
  "LogType":"engine-log",
  "Enabled":false
}'
```

Monitoraggio dell'utilizzo con i parametri di CloudWatch

ElastiCache offre parametri che consentono di monitorare i cluster. Puoi accedere a questi parametri mediante CloudWatch. Per informazioni su CloudWatch, consulta la [documentazione di Amazon CloudWatch](#).

ElastiCache fornisce parametri a livello di host (ad esempio, utilizzo della CPU) e parametri che sono specifici del software del motore di cache (ad esempio, cache gets e cache misses). Questi parametri vengono misurati e pubblicati per ogni nodo di cache in intervalli di 60 secondi.

Important

Valuta se impostare allarmi CloudWatch su determinati parametri chiave, in modo da ricevere una notifica se le prestazioni del cluster di cache iniziano a peggiorare. Per ulteriori informazioni sul tagging, consulta [Quali parametri è opportuno monitorare?](#) in questa guida.

Argomenti

- [Parametri a livello di host](#)
- [Parametri di Redis](#)
- [Quali parametri è opportuno monitorare?](#)
- [Scelta delle statistiche e dei periodi di un parametro](#)
- [Monitoraggio dei parametri del cluster con Amazon CloudWatch](#)

Parametri a livello di host

Il namespace AWS/ElastiCache include i seguenti parametri a livello di host per i singoli nodi di cache. Questi parametri vengono misurati e pubblicati per ogni nodo di cache in intervalli di 60 secondi.

Vedi anche

- [Parametri di Redis](#)

Parametro	Descrizione	Unità
<code>CPUUtilization</code>	La percentuale di utilizzo CPU per l'intero host. Poiché Redis è a thread singolo, consigliamo di monitorare il parametro <code>EngineCPUUtilization</code> per i nodi con almeno 4 vCPU.	Percentuale
<code>CPUCreditBalance</code>	<p>Il numero di crediti CPU ottenuti, che un'istanza ha accumulato da quando è stata lanciata o avviata. Per le T2 Standard <code>CPUCreditBalance</code> include anche il numero di crediti di lancio che sono stati accumulati.</p> <p>I crediti vengono accumulati nel saldo del credito dopo che sono stati ottenuti e rimossi dal saldo del credito una volta spesi. Il saldo del credito ha un limite massimo, determinato dalla dimensione dell'istanza. Una volta che il limite viene raggiunto, tutti i nuovi crediti guadagnati vengono scartati. Per le T2 Standard, i crediti di lancio non contano per il limite.</p> <p>I crediti in <code>CPUCreditBalance</code> sono disponibili affinché l'istanza li spenda per andare oltre l'utilizzo di base della CPU.</p> <p>I parametri di credito CPU sono disponibili solo con una frequenza di 5 minuti.</p> <p>Queste metriche non sono disponibili per le istanze di prestazioni Burstable T2.</p>	Crediti (vCPU/minuti)
<code>CPUCreditUsage</code>	Il numero di crediti CPU spesi dall'istanza per l'utilizzo della CPU. Un credito CPU equivale a un vCPU che viene eseguito al 100% dell'utilizzo per un minuto o una combinazione	Crediti (vCPU/minuti)

Parametro	Descrizione	Unità
	<p>equivalente di vCPU, utilizzo e tempo (per esempio, un vCPU che viene eseguito al 50% dell'utilizzo per due minuti o due vCPU che vengono eseguiti al 25% dell'utilizzo per due minuti).</p> <p>I parametri di credito CPU sono disponibili solo con una frequenza di 5 minuti. Se specifichi un periodo superiore a 5 minuti, usa la statistic a Sum (somma) al posto di quella Average (media).</p> <p>Queste metriche non sono disponibili per le istanze di prestazioni Burstable T2.</p>	
FreeableMemory	La quantità di memoria libera disponibile sull'host. Se puoi liberare spazio dai report OS ciò dipende dalla RAM, dai buffer e dalla cache.	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

Parametro	Descrizione	Unità
NetworkBandwidthInAllowanceExceeded	Il numero di pacchetti accordati o rilasciati perché la larghezza di banda aggregata in ingresso ha superato il valore massimo per l'istanza.	Conteggio
NetworkConntrackAllowanceExceeded	Il numero di pacchetti accordati o rilasciati 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	Il numero di pacchetti accordati o rilasciati perché la larghezza di banda aggregata in uscita ha superato il valore massimo per l'istanza.	Conteggio
NetworkPacketsPerSecondAllowanceExceeded	Numero di pacchetti in coda o rilasciati 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

Parametri di Redis

Il namespace AWS/ElastiCache include i parametri di Redis descritti di seguito.

Ad eccezione di `ReplicationLag` e `EngineCPUUtilization`, questi parametri sono derivati dal comando `info` di Redis. Ogni parametro è calcolato a livello del nodo di cache.

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
<code>ActiveDefragHits</code>	Il numero di riallocazioni di valori al minuto effettuate dal processo di deframmentazione attivo. È derivato dalla statistica <code>active_defrag_hits</code> in Redis INFO .	Numero
<code>AuthenticationFailures</code>	Numero totale di tentativi non riusciti di autenticazione su Redis utilizzando il comando <code>AUTH</code> . 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 assegnati da Redis per tutti gli scopi, incluso il dataset, i buffer ecc.	Byte
<code>BytesUsedForCache</code>	Dimension: <code>Tier=Memory</code> per i cluster Redis che utilizzano Tiering di dati : il numero totale di byte utilizzati per la cache dalla memoria. Questo è il valore di statistica <code>used_memory</code> in Informazioni su Redis .	Byte

Parametro	Descrizione	Unità
	Dimension: Tier=SSD per i cluster Redis che utilizzano Tiering di dati : il numero totale di byte utilizzati per la cache dall'SSD.	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
CacheHits	Il numero di ricerche di chiavi di sola lettura nella directory principale. È derivato dalla statistica <code>keyspace_hits</code> in Redis INFO .	Conteggio
CacheMisses	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
CommandAuthorizationFailures	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

Parametro	Descrizione	Unità
CacheHitRate	Indica l'efficienza di utilizzo dell'istanza Redis. Se il rapporto cache è inferiore a circa 0,8, significa che una quantità significativa di chiavi sono state rimosse, sono scadute o non esistono. Questo viene calcolato utilizzando le statistiche <code>cache_hits</code> e <code>cache_misses</code> nel modo seguente: $\text{cache_hits} / (\text{cache_hits} + \text{cache_misses})$.	Percentuale
ChannelAuthorizationFailures	Numero totale di tentativi non riusciti da parte degli utenti di accedere a canali per i quali non dispongono di autorizzazione di accesso. Ulteriori informazioni sui singoli errori di autenticazione sono disponibili tramite il comando LOG DI CONTROLLO ACCESSI . Consigliamo di impostare un allarme su questa metrica per rilevare i tentativi di accesso non autorizzato.	Conteggio
CurrConnections	Numero di connessioni client, escluse le connessioni dalle repliche di lettura. ElastiCache utilizza da due a quattro di queste connessioni per monitorare il cluster in ogni caso. È derivato dalla statistica <code>connected_clients</code> in Redis INFO .	Conteggio
CurrItems	Il numero di elementi nella cache. È derivato dalla statistica <code>keyspace</code> di Redis sommando tutte le chiavi nell'intero <code>keyspace</code> .	Conteggio
	Dimension: <code>Tier=Memory</code> per cluster che utilizzano Tiering di dati . Il numero di elementi in memoria.	Conteggio

Parametro	Descrizione	Unità
	Dimension: Tier=SSD (Solid State Drive) per cluster che utilizzano Tiering di dati . Il numero di elementi nell'SSD.	Conteggio
CurrVolatileItems	Numero totale di chiavi in tutti i database con un set ttl. È derivato dalla statistica expires di Redis sommando tutte le chiavi con un set ttl nell'intero keyspace.	Conteggio
DatabaseCapacityUsagePercentage	<p>Percentuale della capacità di dati totale disponibile per il cluster in uso. Questa metrica viene calcolata come segue:</p> $\text{used_memory} / \text{maxmemory}$ <p>Nelle istanze dei livelli di dati, la metrica viene calcolata come:</p> $(\text{used_memory} - \text{mem_not_counted_for_evict} + \text{SSD used}) / (\text{maxmemory} + \text{SSD total capacity})$ <p>dove <code>used_memory</code> e <code>maxmemory</code> vengono presi da Redis INFO.</p>	Percentuale

Parametro	Descrizione	Unità
DatabaseCapacityUsageCountedForEvictPercentage	<p>Percentuale della capacità di dati totale per il cluster in uso, esclusa la memoria utilizzata per i costi di gestione e il COB. Questa metrica viene calcolata come segue:</p> $\frac{\text{used_memory} - \text{mem_not_counted_for_evict}}{\text{maxmemory}}$ <p>Nelle istanze dei livelli di dati, la metrica viene calcolata come:</p> $\frac{(\text{used_memory} + \text{SSD used})}{(\text{maxmemory} + \text{SSD total capacity})}$ <p>dove <code>used_memory</code> e <code>maxmemory</code> vengono presi da Redis INFO</p>	Percentuale
DatabaseMemoryUsagePercentage	<p>Percentuale della memoria disponibile per il cluster in uso. Questo viene calcolato utilizzando <code>used_memory/maxmemory</code> da Redis INFO.</p>	Percentuale
DatabaseMemoryUsageCountedForEvictPercentage	<p>Percentuale della memoria disponibile per il cluster in uso, esclusa la memoria utilizzata per l'overhead e il COB. Questo viene calcolato utilizzando <code>used_memory - mem_not_counted_for_evict / maxmemory</code> da Redis INFO.</p>	Percentuale

Parametro	Descrizione	Unità
DB0AverageTTL	Esponde avg_ttl di DBO dalla statistica keypace del comando INFO Redis . Le repliche non fanno scadere le chiavi, ma attendono che siano i nodi primari a far scadere le chiavi. Quando un nodo primario fa scadere una chiave (o la rimuove a causa di LRU), sintetizza un comando DEL, che viene trasmesso a tutte le repliche. Pertanto, DB0AverageTTL è 0 per i nodi di replica, poiché non scadono le chiavi e quindi non tracciano il TTL.	Millisecondi

Parametro	Descrizione	Unità
EngineCPUUtilization	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.	Percentuale

Note


Su un host di ElastiCache, ci sono processi in background che monitorano l'host per fornire un'esperienza di database gestito. 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.

Parametro	Descrizione	Unità
	d. Pertanto, si consiglia di monitorare il parametro <code>CPUUtilization</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>GlobalDatastoreReplicationLag</code>	Questo è il ritardo tra il nodo primario della regione secondaria e il nodo primario della regione primaria. Per Redis abilitato in modalità cluster, il ritardo indica il ritardo massimo tra le partizioni.	Secondi
<code>IamAuthenticationExpirations</code>	Il numero totale di connessioni Redis autenticate da IAM scadute. Ulteriori informazioni su Autenticazione con IAM sono disponibili nella guida per l'utente.	Conteggio
<code>IamAuthenticationThrottling</code>	Il numero totale di richieste Redis AUTH o HELLO autenticate da IAM con throttling. Ulteriori informazioni su Autenticazione con IAM sono disponibili nella guida per l'utente.	Conteggio
<code>IsMaster</code>	Indica se il nodo è il nodo primario della partizione/del cluster corrente. Il parametro può essere 0 (non primario) o 1 (primario).	Conteggio

Parametro	Descrizione	Unità
KeyAuthorizationFailures	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
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 la memorizzazione nella cache lato client e notifica ai client quando le chiavi vengono modificate.	Conteggio
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

Parametro	Descrizione	Unità
NewConnections	<p>Il numero totale di connessioni accettate dal server durante questo periodo. È derivato dalla statistica <code>total_connections_received</code> in Redis INFO.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>Se si utilizza ElastiCache per Redis versione 5 o precedenti, ElastiCache utilizza da due a quattro delle connessioni segnalate da questa metrica per monitorare il cluster. Tuttavia, quando si utilizza ElastiCache per Redis versione 6 o successive, le connessioni utilizzate da ElastiCache per monitorare il cluster non sono incluse in questa metrica.</p> </div>	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
MasterLinkHealthStatus	Questo stato ha due valori: 0 e 1. Il valore 0 indica che i dati nel nodo primario di ElastiCache non sono sincronizzati con Redis su EC2. Il valore 1 indica che i dati sono in fase di sincronizzazione. Per completare la migrazione, utilizzare l'operazione dell'API CompleteMigration .	Boolean

Parametro	Descrizione	Unità
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. Questo parametro è rappresentativo del carico di scrittura sul gruppo di replica. È derivato dalla statistica <code>master_repl_offset</code> in Redis INFO .	Byte
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. Per motore Redis versione 5.0.6 e successive, il ritardo può essere misurato in millisecondi.	Secondi
SaveInProgress	Questo parametro binario restituisce 1 ogni volta che è in corso un salvataggio in background (con o senza fork), altrimenti restituisce 0. Un processo di salvataggio in background è in genere utilizzato durante snapshot e sincronizzazioni. Queste operazioni possono peggiorare le prestazioni. Utilizzando il parametro <code>SaveInProgress</code> , è possibile determinare se il peggioramento delle prestazioni è dovuto o meno a un processo di salvataggio in background. È derivato dalla statistica <code>rdb_bgsave_in_progress</code> in Redis INFO .	Boolean

Parametro	Descrizione	Unità
TrafficManagementActive	<p>Indica se ElastiCache per Redis sta gestendo attivamente il traffico regolando il traffico allocato ai comandi, al monitoraggio o alla replica in ingresso. Il traffico viene gestito quando vengono inviati al nodo più comandi di quanti ne possano essere elaborati da Redis e viene utilizzato per mantenere la stabilità e il funzionamento ottimale del motore. Eventuali punti dati di 1 possono indicare che il nodo è sottodimensionato per il carico di lavoro fornito.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Se questa metrica rimane attiva, valuta il cluster per decidere se è necessario aumentare o aumentare orizzontalmente. Le metriche correlate includono NetworkBandwidthOutputAllowanceExceeded e EngineCPUUtilization .</p> </div>	Boolean

Disponibilità di EngineCPUUtilization

Le regioni AWS elencate di seguito sono disponibili in tutti i tipi di nodo supportati.

Regione	Nome Regione
us-east-2	Stati Uniti orientali (Ohio)
us-east-1	Stati Uniti orientali (Virginia settentrionale)
us-west-1	Stati Uniti occidentali (California settentrionale)
us-west-2	Stati Uniti occidentali (Oregon)

Regione	Nome Regione
ap-northeast-1	Asia Pacifico (Tokyo)
ap-northeast-2	Asia Pacifico (Seul)
ap-northeast-3	Asia Pacifico (Osaka-Locale)
ap-east-1	Asia Pacifico (Hong Kong)
ap-south-1	Asia Pacifico (Mumbai)
ap-southeast-1	Asia Pacifico (Singapore)
ap-southeast-2	Asia Pacifico (Sydney)
ap-southeast-3	Asia Pacifico (Giacarta)
ca-central-1	Canada (Centrale)
cn-north-1	Cina (Pechino)
cn-northwest-2	Cina (Ningxia)
me-south-1	Medio Oriente (Bahrein)
eu-central-1	Europa (Francoforte)
eu-west-1	Europa (Irlanda)
eu-west-2	Europa (Londra)
eu-west-3	UE (Parigi)
eu-south-1	Europa (Milano)
af-south-1	Africa (Città del Capo)
eu-north-1	Europa (Stoccolma)
sa-east-1	Sud America (São Paulo)

Regione	Nome Regione
us-gov-west-1	AWS GovCloud (US-West)
us-gov-east-1	AWS GovCloud (US-East)

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, il tempo totale della CPU utilizzato da questi comandi e la CPU media consumata per esecuzione dei comandi. Per ogni tipo di comando, viene aggiunta la seguente riga: `cmdstat_XXX: calls=XXX,usec=XXX,usec_per_call=XXX`.

I parametri di latenza elencati di seguito sono calcolati utilizzando le statistiche `commandstats` da [Redis INFO](#). Sono calcolati nel modo seguente: $\text{delta}(\text{usec})/\text{delta}(\text{calls})$. `delta` viene calcolato come differenza entro un minuto. La latenza è definita come tempo CPU impiegato da ElastiCache per elaborare il comando. Si noti che per i cluster che utilizzano il tiering di dati, il tempo necessario per recuperare gli elementi dall'SSD non è incluso in queste misurazioni.

Per l'elenco completo dei comandi disponibili, consulta [comandi Redis](#) nella documentazione relativa a Redis.

Parametro	Descrizione	Unità
<code>ClusterBasedCmds</code>	Il numero totale di comandi basati su cluster. È ricavato dalla statistica <code>commandstats</code> di Redis sommando tutti i comandi che agiscono in base a un cluster (<code>cluster slot</code> , <code>cluster info</code> e così via).	Conteggio
<code>ClusterBasedCmdsLatency</code>	Latenza dei comandi basati su cluster.	Microsecondi
<code>EvalBasedCmds</code>	Numero totale di comandi per i comandi basati su valutazione. Questo è derivato dalla <code>commandstats</code> statistica sommando <code>eval</code> , <code>evalsha</code> di Redis.	Conteggio
<code>EvalBasedCmdsLatency</code>	Latenza dei comandi basati su EVAL.	Microsecondi

Parametro	Descrizione	Unità
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
GeoSpatialBasedCmdsLatency	Latenza dei comandi basati su Geospaziali.	Microsecondi
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
GetTypeCmdsLatency	Latenza dei comandi di lettura.	Microsecondi
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
HashBasedCmdsLatency	Latenza dei comandi basati su hash.	Microsecondi
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
HyperLogLogBasedCmdsLatency	Latenza dei comandi HyperLogLogBased.	Microsecondi

Parametro	Descrizione	Unità
JsonBasedCmds	Il numero totale di comandi JSON, inclusi i comandi di lettura e scrittura. È derivato dalla statistica <code>commandstats</code> di Redis sommando tutti i comandi JSON che agiscono sulle chiavi JSON.	Conteggio
JsonBasedCmdsLatency	Latenza di tutti i comandi JSON, inclusi i comandi di lettura e scrittura.	Microsecondi
JsonBasedGetCmds	Il numero totale di comandi JSON di sola lettura. È derivato dalla statistica <code>commandstats</code> di Redis sommando tutti i comandi di lettura JSON che agiscono sulle chiavi JSON.	Conteggio
JsonBasedGetCmdsLatency	Latenza dei comandi di sola lettura JSON.	Microsecondi
JsonBasedSetCmds	Il numero totale di comandi di scrittura JSON. È derivato dalla statistica <code>commandstats</code> di Redis sommando tutti i comandi di scrittura JSON che agiscono sulle chiavi JSON.	Conteggio
JsonBasedSetCmdsLatency	Latenza dei comandi di scrittura JSON.	Microsecondi
KeyBasedCmds	Il numero totale di comandi basati su chiavi. È derivato dalla statistica <code>commandstats</code> di Redis sommando tutti i comandi che agiscono su una o più chiavi nelle strutture di dati multiple (del, expire, rename ecc.).	Conteggio
KeyBasedCmdsLatency	Latenza dei comandi basati su chiave.	Microsecondi

Parametro	Descrizione	Unità
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 (<code>lindex</code> , <code>lrange</code> , <code>lpush</code> , <code>ltrim</code> , etc).	Conteggio
ListBasedCmdsLatency	Latenza dei comandi basati su elenchi.	Microsecondi
NonKeyTypeCmds	Il numero totale di comandi non basati su chiavi. È derivato dalla statistica <code>commandstats</code> di Redis sommando tutti i comandi che non agiscono su una o più chiavi, ad esempio, <code>acl</code> , <code>dbsize</code> o <code>info</code> .	Conteggio
NonKeyTypeCmdsLatency	Latenza dei comandi non basati su chiave.	Microsecondi
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>punsubscribe</code> , <code>publish</code> , <code>pubsub</code> , <code>punsubscribe</code> , <code>ssubscribe</code> , <code>sunsubscribe</code> , <code>spublish</code> , <code>subscribe</code> e <code>unsubscribe</code> .	Conteggio
PubSubBasedCmdsLatency	Latenza dei comandi pub/sub-based.	Microsecondi
SetBasedCmds	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
SetBasedCmdsLatency	Latenza dei comandi basati su set.	Microsecondi

Parametro	Descrizione	Unità
SetTypeCmds	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
SetTypeCmdsLatency	Latenza dei comandi di scrittura.	Microsecondi
SortedSetBasedCmds	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
SortedSetBasedCmdsLatency	Latenza dei comandi basati su ordinazione.	Microsecondi
StringBasedCmds	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
StringBasedCmdsLatency	Latenza dei comandi basati su stringhe.	Microsecondi
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
StreamBasedCmdsLatency	Latenza dei comandi basati sul flusso.	Microsecondi

Quali parametri è opportuno monitorare?

I parametri CloudWatch seguenti offrono informazioni dettagliate riguardo alle prestazioni di ElastiCache. Nella maggior parte dei casi, ti consigliamo di impostare allarmi CloudWatch per questi parametri in modo da poter apportare delle correzioni prima che si verifichino problemi di prestazioni.

Parametri da monitorare

- [CPUUtilization](#)
- [EngineCPUUtilization](#)
- [SwapUsage](#)
- [Espulsioni](#)
- [CurrConnections](#)
- [Memory \(Memoria\)](#)
- [Rete](#)
- [Latenza](#)
- [Replica](#)
- [Gestione del traffico](#)

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%.

Occorre determinare la propria soglia, in base al numero di core nel nodo di cache in uso. Se questa soglia viene superata, e il carico di lavoro principale è da richieste di lettura, dimensiona orizzontalmente il cluster di cache aggiungendo repliche di lettura. Se il carico di lavoro principale è da richieste di scrittura, a seconda della configurazione cluster, ti consigliamo di procedere come segue:

- Cluster di Redis (modalità cluster disabilitata): dimensionare verso l'alto utilizzando un tipo di istanza di cache più grande.
- Cluster di Redis (modalità cluster disabilitata): aggiungi più partizioni per distribuire il carico di lavoro tra più nodi primari.

Tip

Anziché utilizzare il parametro a livello di host `CPUUtilization`, gli utenti Redis possono utilizzare il parametro `Redis EngineCPUUtilization` che indica la percentuale di utilizzo sul core del motore Redis. Per vedere se questo parametro è disponibile sui nodi e per ulteriori informazioni, consulta la sezione relativa ai [parametri per Redis](#).

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 questo parametro è disponibile sui nodi e per ulteriori informazioni, consulta la sezione relativa ai [parametri per Redis](#).

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 questo parametro è disponibile sui nodi e per ulteriori informazioni, consulta la sezione relativa ai [parametri per Redis](#).

Per ulteriori informazioni, consulta la sezione CPU su [Monitoraggio delle best practice con Amazon ElastiCache for Redis utilizzando Amazon CloudWatch](#).

SwapUsage

Si tratta di un parametro a livello di host restituito in byte. Per ulteriori informazioni, consulta [Parametri a livello di host](#).

La metrica `FreeableMemory` CloudWatch è vicina a 0 (cioè inferiore a 100 MB) o la metrica `SwapUsage` maggiore della metrica `FreeableMemory` indica che un nodo è sotto pressione di memoria. In tal caso, consulta i seguenti argomenti:

- [Verifica della disponibilità di memoria sufficiente per creare uno snapshot Redis](#)
- [Gestione della memoria prenotata](#)

Espulsioni

Si tratta di un parametro del motore di cache. Ti consigliamo di determinare la tua soglia di allarme per questo parametro in base alle esigenze dell'applicazione.

CurrConnections

Si tratta di un parametro del motore di cache. Ti consigliamo di determinare la tua soglia di allarme per questo parametro in base alle esigenze dell'applicazione.

Un numero crescente di CurrConnections potrebbe indicare un problema con l'applicazione; per risolvere questo problema è opportuno esaminare il comportamento dell'applicazione.

Per ulteriori informazioni, consulta la sezione Connessioni su [Monitoraggio delle best practice con Amazon ElastiCache for Redis utilizzando Amazon CloudWatch](#).

Memory (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](#).

Per ulteriori informazioni, consulta la sezione Memoria su [Monitoraggio delle best practice con Amazon ElastiCache for Redis utilizzando Amazon CloudWatch](#).

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 nodo, consulta [Prezzi di Amazon ElastiCache](#).

Per ulteriori informazioni, consulta la sezione Rete su [Monitoraggio delle best practice con Amazon ElastiCache for Redis utilizzando Amazon CloudWatch](#).

Latenza

È possibile misurare la latenza di un comando con un set di parametri CloudWatch che forniscono latenze aggregate per struttura di dati. Queste metriche di latenza sono calcolate utilizzando la statistica `commandstats` dal comando di Redis [INFO](#).

Per ulteriori informazioni, consulta la sezione Latenza su [Monitoraggio delle best practice con Amazon ElastiCache for Redis utilizzando Amazon CloudWatch](#).

Replica

Il volume dei dati da replicare è visibile tramite il parametro `ReplicationBytes`. Sebbene questo parametro sia rappresentativo del carico di scrittura sul gruppo di replica, non fornisce informazioni dettagliate sull'integrità della replica. A tale scopo, è possibile utilizzare il parametro `ReplicationLag`.

Per ulteriori informazioni, consulta la sezione Replica su [Monitoraggio delle best practice con Amazon ElastiCache for Redis utilizzando Amazon CloudWatch](#).

Gestione del traffico

ElastiCache per Redis gestisce automaticamente il traffico verso un nodo quando vengono inviati al nodo più comandi in ingresso di quanti ne possano essere elaborati da Redis. Ciò viene fatto per mantenere il funzionamento e la stabilità ottimali del motore.

Quando il traffico viene gestito attivamente su un nodo, la metrica `TrafficManagementActive` emetterà punti dati pari a 1. Ciò indica che il nodo è sottodimensionato per il carico di lavoro fornito. Se questa metrica resta 1 per lunghi periodi di tempo, valuta il cluster per decidere se è necessario aumentare o applicare la scalabilità orizzontale.

Per ulteriori informazioni, consulta la metrica `TrafficManagementActive` nella pagina [Metriche](#).

Scelta delle statistiche e dei periodi di un parametro

Sebbene CloudWatch consenta di scegliere eventuali statistiche e periodi per ogni parametro, 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 i campioni di ElastiCache vengono pubblicati per una durata di 60 secondi per ogni singolo nodo di cache. Per qualsiasi periodo di 60 secondi, un parametro del nodo di cache conterrà solo un singolo campione.

Per ulteriori informazioni su come recuperare parametri per i nodi di cache, consulta [Monitoraggio dei parametri del cluster con Amazon CloudWatch](#).

Monitoraggio dei parametri del cluster con Amazon CloudWatch

ElastiCache e CloudWatch sono integrati per consentire di acquisire vari parametri. Puoi monitorare questi parametri utilizzando CloudWatch.

Note

Il seguente esempio di interfaccia riga di comando richiede gli strumenti a riga di comando CloudWatch. Per ulteriori informazioni su CloudWatch e per scaricare gli strumenti per sviluppatori, consulta la pagina del prodotto CloudWatch.

Le procedure seguenti illustrano come utilizzare CloudWatch per acquisire statistiche sullo spazio di archiviazione di un cluster di cache nell'ultima ora.

Note

I valori StartTime ed EndTime degli esempi seguenti sono forniti solo a scopo illustrativo. Dovranno quindi, all'occorrenza, essere sostituiti con gli orari di inizio e fine appropriati per i nodi di cache.

Per informazioni sui limiti ElastiCache, consulta [AWS Restrizioni dei servizi](#) per ElastiCache.

Monitoraggio dei parametri di CloudWatch Cluster e Node (console)

Per visualizzare le statistiche di utilizzo della CPU relative a un cluster di cache

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Scegliere i nodi di cache di cui visualizzare i parametri.

Note

La selezione di oltre 20 nodi disabilita la visualizzazione dei parametri sulla console.

- a. Nella pagina Cache Clusters (Cluster di cache) della console di gestione di AWS, fare clic sul nome di uno o più cluster di cache.

Viene visualizzata la pagina con i dettagli del cluster di cache.

- b. Fare clic sulla scheda Nodes (Nodi) nella parte superiore della finestra.
- c. Scegliere i nodi di cache di cui visualizzare i parametri nella scheda Nodes (Nodi) della finestra dei dettagli.

Nella parte inferiore della finestra della console viene visualizzato l'elenco dei parametri CloudWatch disponibili.

- d. Fare clic sul parametro CPU Utilization (Utilizzo CPU).

Si apre la console CloudWatch, che mostra i parametri selezionati. 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 metriche del cluster e dei nodi CloudWatch tramite la CLI

Per visualizzare le statistiche di utilizzo della CPU relative a un cluster di cache

- Per Linux, macOS o Unix:

```
aws cloudwatch get-metric-statistics \  
  --namespace AWS/ElastiCache \  
  --metric-name CPUUtilization \  
  --dimensions='[{"Name":"CacheClusterId","Value":"test"},  
{"Name":"CacheNodeId","Value":"0001"}]' \  
  --statistics=Average \  
  --start-time 2018-07-05T00:00:00 \  
  --end-time 2018-07-06T00:00:00 \  

```

```
--period=3600
```

Per Windows:

```
aws cloudwatch get-metric-statistics ^  
  --namespace AWS/ElastiCache ^  
  --metric-name CPUUtilization ^  
  --dimensions='[{"Name":"CacheClusterId","Value":"test"},  
{"Name":"CacheNodeId","Value":"0001"}]'^ ^  
  --statistics=Average ^  
  --start-time 2018-07-05T00:00:00 ^  
  --end-time 2018-07-06T00:00:00 ^  
  --period=3600
```

Monitoraggio delle metriche del cluster e dei nodi CloudWatch tramite l'API CloudWatch

Per visualizzare le statistiche di utilizzo della CPU relative a un cluster di cache

- Chiamare l'API di CloudWatch `GetMetricStatistics` con i parametri riportati di seguito (tenendo presente che gli orari di inizio e fine sono puramente esemplificativi e dovranno, all'occorrenza, essere sostituiti con quelli appropriati):
 - `Statistics.member.1=Average`
 - `Namespace=AWS/ElastiCache`
 - `StartTime=2013-07-05T00:00:00`
 - `EndTime=2013-07-06T00:00:00`
 - `Period=60`
 - `MeasureName=CPUUtilization`
 - `Dimensions=CacheClusterId=mycachecluster,CacheNodeId=0002`

Example

```
http://monitoring.amazonaws.com/  
  ?Action=GetMetricStatistics  
  &SignatureVersion=4  
  &Version=2014-12-01  
  &StartTime=2018-07-05T00:00:00
```

```
&EndTime=2018-07-06T23:59:00
&Period=3600
&Statistics.member.1=Average
&Dimensions.member.1="CacheClusterId=mycachecluster"
&Dimensions.member.2="CacheNodeId=0002"
&Namespace=&AWS;/ElastiCache
&MeasureName=CPUUtilization
&Timestamp=2018-07-07T17%3A48%3A21.746Z
&AWS;AccessKeyId=<&AWS; Access Key ID>
&Signature=<Signature>
```

Monitoraggio di Amazon SNS per gli eventi ElastiCache

Quando in un cluster accadono eventi significativi, ElastiCache invia una notifica a un argomento Amazon SNS specifico. 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 ElastiCache Amazon SNS](#)
- [Visualizzazione di eventi ElastiCache C](#)
- [Notifiche di eventi Amazon SNS](#)

Gestione ElastiCache Amazon SNS

Puoi configurare ElastiCache in modo tale che invii notifiche sugli eventi importanti del cluster con 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 sezioni che seguono illustrano come aggiungere un argomento Amazon SNS con la console AWS, la AWS CLI o l'API di ElastiCache.

Aggiunta di un argomento Amazon SNS (console)

Nella seguente procedura viene mostrato come aggiungere un argomento Amazon SNS a un cluster. Per aggiungere un argomento Amazon SNS a un gruppo di replica, basta scegliere tale gruppo anziché un cluster nella fase 2; per il resto, la procedura è la stessa.

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. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
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.

Aggiunta di un argomento Amazon SNS (AWS CLI)

Per aggiungere o modificare l'argomento Amazon SNS di un cluster, utilizzare il comando AWS CLI del `modify-cache-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 elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --notification-topic-arn arn:aws:sns:us-  
west-2:123456789xxx:ElastiCacheNotifications
```

Per Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --notification-topic-arn arn:aws:sns:us-west-2:123456789xx:ElastiCacheNotifications
```

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

Aggiunta di un argomento Amazon SNS (API ElastiCache)

Per aggiungere o modificare l'argomento Amazon SNS di un cluster, occorre chiamare l'operazione `ModifyCacheCluster` con i seguenti parametri:

- `CacheClusterId=my-cluster`
- `TopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications`

Example

```
https://elasticache.amazon.com/  
  ?Action=ModifyCacheCluster  
  &ApplyImmediately=false  
  &CacheClusterId=my-cluster  
  &NotificationTopicArn=arn%3Aaws%3Asns%3Aus-  
west-2%3A565419523791%3AElastiCacheNotifications  
  &Version=2014-12-01  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

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

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 con la AWS Management Console

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Per visualizzare l'elenco dei cluster che eseguono Redis, scegliere Redis (Redis) nel pannello di navigazione.
3. Scegliere la casella a sinistra del cluster di cui modificare le notifiche.
4. Scegli Modifica.
5. In Modify Cluster (Modifica cluster) nella sezione Topic for SNS Notification (Argomento per notifica SNS), scegliere Disable Notifications (Disabilita notifiche).
6. Scegli Modifica.

Attivazione e disattivazione delle notifiche Amazon SNS (AWS CLI)

Per disabilitare le notifiche Amazon SNS, occorre utilizzare il comando `modify-cache-cluster` con i seguenti parametri:

Per Linux, macOS o Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --notification-topic-status inactive
```

Per Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --notification-topic-status inactive
```

Attivazione e disattivazione delle notifiche Amazon SNS (API ElastiCache)

Per disabilitare le notifiche Amazon SNS, occorre chiamare l'operazione `ModifyCacheCluster` con i seguenti parametri:

- `CacheClusterId=my-cluster`
- `NotificationTopicStatus=inactive`

Questa chiamata restituisce un output simile al seguente:

Example

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheCluster  
&ApplyImmediately=false  
&CacheClusterId=my-cluster  
&NotificationTopicStatus=inactive  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```


Visualizzazione di eventi ElastiCache C

ElastiCache registra gli eventi correlati alle istanze, nonché ai gruppi di sicurezza e di parametri del cluster. Queste informazioni includono la data, l'ora, il nome e tipo di fonte e una descrizione dell'evento. Gli eventi possono essere facilmente recuperati dal registro utilizzando la console di ElastiCache, il comando `describe-events` di AWS CLI o l'operazione API `DescribeEvents` di ElastiCache.

Le procedure seguenti mostrano come visualizzare tutti gli eventi ElastiCache delle ultime 24 ore (1440 minuti).

Visualizzazione degli eventi ElastiCache (Console)

La seguente procedura consente di visualizzare eventi con la console ElastiCache.

Per visualizzare eventi utilizzando la console ElastiCache

1. Accedere alla AWS Management Console e aprire la console ElastiCache all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Per l'elenco di tutti gli eventi disponibili, scegliere Events (Eventi) nel pannello di navigazione.

Nella schermata Eventi, ogni riga dell'elenco rappresenta un evento e di quest'ultimo riporta l'fonte, il tipo (`cache-cluster`, `cache-parameter-group`, `cache-security-group` o `cache-subnet-group`), l'orario nel fuso GMT e la descrizione.

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 ElastiCache (AWS CLI)

Per generare una lista di eventi ElastiCache usando la AWS CLI, utilizzare il 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 di cache.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

Il codice seguente elenca tutti gli eventi delle ultime 24 ore (1440 minuti).

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

L'output del comando `describe-events` è simile a quello riportato.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
{
  "Events": [
    {
      "SourceIdentifier": "my-mem-cluster",
      "SourceType": "cache-cluster",
      "Message": "Finished modifying number of nodes from 1 to 3",
      "Date": "2020-06-09T02:01:21.772Z"
    },
    {
      "SourceIdentifier": "my-mem-cluster",
      "SourceType": "cache-cluster",
      "Message": "Added cache node 0002 in availability zone us-west-2a",
      "Date": "2020-06-09T02:01:21.716Z"
    },
    {
      "SourceIdentifier": "my-mem-cluster",
      "SourceType": "cache-cluster",
      "Message": "Added cache node 0003 in availability zone us-west-2a",
      "Date": "2020-06-09T02:01:21.706Z"
    },
    {
      "SourceIdentifier": "my-mem-cluster",
      "SourceType": "cache-cluster",
      "Message": "Increasing number of requested nodes",
      "Date": "2020-06-09T01:58:34.178Z"
    },
    {
      "SourceIdentifier": "mycluster-0003-004",
      "SourceType": "cache-cluster",
      "Message": "Added cache node 0001 in availability zone us-west-2c",
      "Date": "2020-06-09T01:51:14.120Z"
    },
    {
      "SourceIdentifier": "mycluster-0003-004",
      "SourceType": "cache-cluster",
      "Message": "This cache cluster does not support persistence (ex:
      'appendonly'). Please use a different instance type to enable persistence.",
      "Date": "2020-06-09T01:51:14.095Z"
    }
  ]
}
```

```
  },
  {
    "SourceIdentifier": "mycluster-0003-004",
    "SourceType": "cache-cluster",
    "Message": "Cache cluster created",
    "Date": "2020-06-09T01:51:14.094Z"
  },
  {
    "SourceIdentifier": "mycluster-0001-005",
    "SourceType": "cache-cluster",
    "Message": "Added cache node 0001 in availability zone us-west-2b",
    "Date": "2020-06-09T01:42:55.603Z"
  },
  {
    "SourceIdentifier": "mycluster-0001-005",
    "SourceType": "cache-cluster",
    "Message": "This cache cluster does not support persistence (ex:
'appendonly'). Please use a different instance type to enable persistence.",
    "Date": "2020-06-09T01:42:55.576Z"
  },
  {
    "SourceIdentifier": "mycluster-0001-005",
    "SourceType": "cache-cluster",
    "Message": "Cache cluster created",
    "Date": "2020-06-09T01:42:55.574Z"
  },
  {
    "SourceIdentifier": "mycluster-0001-004",
    "SourceType": "cache-cluster",
    "Message": "Added cache node 0001 in availability zone us-west-2b",
    "Date": "2020-06-09T01:28:40.798Z"
  },
  {
    "SourceIdentifier": "mycluster-0001-004",
    "SourceType": "cache-cluster",
    "Message": "This cache cluster does not support persistence (ex:
'appendonly'). Please use a different instance type to enable persistence.",
    "Date": "2020-06-09T01:28:40.775Z"
  },
  {
    "SourceIdentifier": "mycluster-0001-004",
    "SourceType": "cache-cluster",
    "Message": "Cache cluster created",
    "Date": "2020-06-09T01:28:40.773Z"
  }
}
```

```
    }  
  ]  
}
```

Per ulteriori informazioni, tra cui i parametri disponibili e i valori consentiti per tali parametri, consulta [describe-events](#).

Visualizzazione degli eventi ElastiCache (API ElastiCache)

Per generare una lista di eventi ElastiCache usando l' API di ElastiCache , utilizzare l'operazione 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 cache-cluster più recenti.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeEvents  
&MaxRecords=40  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&SourceType=cache-cluster  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

Il codice seguente elenca gli eventi cache-cluster delle ultime 24 ore (1440 minuti).

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeEvents  
&Duration=1440  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&SourceType=cache-cluster  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

Le operazioni descritte in precedenza dovrebbero generare un output simile al seguente.

```
<DescribeEventsResponse xmlns="http://elasticache.amazonaws.com/doc/2015-02-02/">  
  <DescribeEventsResult>
```

```
<Events>
  <Event>
    <Message>Cache cluster created</Message>
    <SourceType>cache-cluster</SourceType>
    <Date>2015-02-02T18:22:18.202Z</Date>
    <SourceIdentifier>mem01</SourceIdentifier>
  </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

Quando si verificano eventi significativi in un cluster di cache, ElastiCache può inviare messaggi tramite Amazon Simple Notification Service (SNS). Questa caratteristica può essere utilizzata per aggiornare gli elenchi dei server sulle macchine client connesse a endpoint di singoli nodi di cache afferenti a un cluster di cache.

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 ElastiCache.
- L'account AWS titolare dell'argomento Amazon SNS deve corrispondere al proprietario del cluster di cache su cui sono abilitate le notifiche.
- L'argomento Amazon SNS che si sta pubblicando non può essere crittografato.

Note

È possibile allegare un argomento di Amazon SNS crittografato (a riposo) al cluster. Tuttavia, lo stato dell'argomento dalla console ElastiCache verrà visualizzato come inattivo, il che disassocia efficacemente l'argomento dal cluster quando ElastiCache invia i messaggi all'argomento.


- L'argomento Amazon SNS deve trovarsi nella stessa regione del cluster ElastiCache.


ElastiCache

I seguenti eventi ElastiCache attivano notifiche Amazon SNS. Per ulteriori informazioni sui dettagli degli eventi, consulta [Visualizzazione di eventi ElastiCache C](#).

Nome evento	Messaggio	Descrizione
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	Un nodo di cache è stato aggiunto al cluster di cache ed è pronto per l'uso.
ElastiCache:AddCacheNodeFailed due to insufficient free IP addresses	ElastiCache:AddCacheNodeFailed : <i>cluster-name</i>	Non è stato possibile aggiungere un nodo di cache per l'insufficienza di indirizzi IP disponibili.
ElastiCache:CacheClusterParametersChanged	ElastiCache:CacheClusterParametersChanged : <i>cluster-name</i>	Uno o più parametri del cluster di cache sono stati modificati.
ElastiCache:cacheclusterProvisioningComplete	ElastiCache:CacheClusterProvisioningComplete <i>cluster-name-0001-005</i>	Il provisioning di un cluster di cache è completo e i nodi di cache nel cluster sono pronti all'uso.
ElastiCache:CacheClusterProvisioningFailed due to incompatible network state	ElastiCache:CacheClusterProvisioning	È stato effettuato un tentativo di avvio di un nuovo cluster di cache in un cloud privato

Nome evento	Messaggio	Descrizione
	Failed : <i>cluster-name</i>	virtuale (VPC, Virtual Private Cloud) inesistente.
ElastiCache:CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	Dimensionamento di un cluster di cache completato.
Elasticache: cacheclusterScalingFailed	ElasticaCacheclusterScalingFailed: <i>nome-cluster</i>	Non è riuscita l'operazione di dimensionamento di un cluster di cache.
Elasticache:cacheclusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	<p>Si è verificato uno dei seguenti eventi:</p> <ul style="list-style-type: none"> • L'elenco dei gruppi di sicurezza cache autorizzati per il cluster di cache è stato modificato. • Uno o più nuovi gruppi di sicurezza EC2 sono stati ammessi nei gruppi di sicurezza cache associati al cluster di cache. • Uno o più gruppi di sicurezza EC2 sono stati rimossi dai gruppi di sicurezza cache associati al cluster di cache.

Nome evento	Messaggio	Descrizione
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	<p>ElastiCache ha rilevato che l'host che esegue un nodo di cache è danneggiato o irraggiungibile ed è iniziata la sostituzione di tale nodo.</p> <div data-bbox="1068 495 1507 810"><p> Note</p><p>La voce DNS relativa al nodo di cache in sostituzione non cambia.</p></div> <p>Nella maggior parte dei casi, non è necessario aggiornare l'elenco dei server per i client, quando si verifica questo evento. Tuttavia, alcune librerie client della cache potrebbero smettere di utilizzare il nodo di cache nonostante la sostituzione da parte di ElastiCache; in tal caso, l'applicazione dovrà aggiornare l'elenco dei server.</p>

Nome evento	Messaggio	Descrizione
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	<p>ElastiCache ha rilevato che l'host che esegue un nodo di cache è danneggiato o irraggiungibile ed effettuato la sostituzione di tale nodo.</p> <div data-bbox="1068 495 1507 810" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note</p> <p>La voce DNS relativa al nodo di cache in sostituzione non cambia.</p> </div> <p>Nella maggior parte dei casi, non è necessario aggiornare l'elenco dei server per i client, quando si verifica questo evento. Tuttavia, alcune librerie client della cache potrebbero smettere di utilizzare il nodo di cache nonostante la sostituzione da parte di ElastiCache; in tal caso, l'applicazione dovrà aggiornare l'elenco dei server.</p>
ElastiCache:cacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	<p>Uno o più nodi di cache sono stati riavviati.</p> <p>Messaggio (Memcached): "Cache node %s shutdown" Secondo messaggio: "Cache node %s restarted"</p>

Nome evento	Messaggio	Descrizione
ElastiCache: Certifica terenewalComplete (solo Redis)	ElastiCache: Certif icateRenewalComple te	Il certificato Amazon CA è stato rinnovato correttamente.
ElastiCache:CreateReplicati onGroupComplete	ElastiCache:Create ReplicationGroupCo mplete : <i>cluster-n ame</i>	Il gruppo di replica è stato creato correttamente.
ElastiCache>DeleteCacheClus terComplete	ElastiCache>Delete CacheClusterComple te : <i>cluster-name</i>	È stata completata la procedura di eliminazione di un cluster di cache con tutti i nodi associati.
ElastiCache: FailoverC omplete (solo Redis)	ElastiCache:Failov erComplete : <i>mycluster</i>	Il failover su un nodo di replica ha avuto esito positivo.
ElastiCache:ReplicationGrou pIncreaseReplicaCountFinish ed	ElastiCache:Replic ationGroupIncrease ReplicaCountFinish ed : <i>cluster-n ame-0001-005</i>	Il numero di repliche nel cluster è stato aumentato.
Elasticache:ReplicationGrou pIncreaseReplicaCount	ElastiCache:Replic ationGroupIncrease ReplicaCountStarte d : <i>cluster-n ame-0003-004</i>	Il processo di aggiunta di repliche al cluster è iniziato.
Elasticache:NoderePlacement Canceled	ElastiCache:NodeRe placementCanceled : <i>cluster-name</i>	È stata annullata la sostituzi one programmata di un nodo nel cluster.

Nome evento	Messaggio	Descrizione
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	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 .
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	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 .
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Un nodo di cache è stato rimosso dal cluster di cache.
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	Operazione di scalabilità verticale sul gruppo di replica completata correttamente.
ElastiCache:ReplicationGroupScalingFailed	"Failed applying modification to cache node type to %s."	Operazione di scalabilità verticale sul gruppo di replica non riuscita.
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	È disponibile un aggiornamento self-service per il nodo.

Nome evento	Messaggio	Descrizione
ElastiCache: SnapshotComplete (solo Redis)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	È stata correttamente acquisita una snapshot di cache.
ElastiCache: SnapshotFailed (solo Redis)	SnapshotFailed : <i>cluster-name</i>	Non è stata portata a compimento l'acquisizione di una snapshot di cache. Per informazioni dettagliate sulle cause, consulta gli eventi della cache del cluster. Lo stato della snapshot, riportato in DescribeSnapshots , sarà failed.

Argomenti correlati

- [Visualizzazione di eventi ElastiCache C](#)

Registrazione delle chiamate all'API Amazon ElastiCache con AWS CloudTrail

Amazon ElastiCache è integrato con AWS CloudTrail, un servizio che offre un registro ecord delle operazioni eseguite da un utente, un ruolo o un servizio AWS in Amazon ElastiCache. CloudTrail acquisisce tutte le chiamate API per Amazon ElastiCache come eventi, incluse le chiamate dalla console Amazon ElastiCache e dal codice alle operazioni API di Amazon ElastiCache. Se crei un percorso, puoi abilitare la distribuzione continua di eventi CloudTrail in un bucket Amazon S3, inclusi gli eventi per Amazon ElastiCache. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella console di CloudTrail in Cronologia eventi. Le informazioni raccolte da CloudTrail consentono di determinare la richiesta effettuata ad Amazon ElastiCache, l'indirizzo IP da cui è partita la richiesta, l'autore della richiesta, il momento in cui è stata eseguita e altri dettagli.

Per ulteriori informazioni su CloudTrail, consultare la [AWS CloudTrail Guida per l'utente di](#).

Informazioni su Amazon ElastiCache in CloudTrail

CloudTrail è abilitato sull'account AWS al momento della sua creazione. Quando si verifica un'attività in Amazon ElastiCache, questa viene registrata in un evento CloudTrail insieme ad altri eventi del servizio AWS nella cronologia degli eventi. È possibile visualizzare, cercare e scaricare gli eventi recenti nell'account AWS. Per ulteriori informazioni, consulta [Visualizzazione di eventi nella cronologia degli eventi di CloudTrail](#).

Per una registrazione continua degli eventi nell'account AWS, inclusi gli eventi per Amazon ElastiCache, crea un percorso. Un percorso abilita la distribuzione da parte di CloudTrail dei 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 percorso 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 servizi AWS per analizzare con maggiore dettaglio e usare i dati evento raccolti nei log CloudTrail. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Panoramica della creazione di un percorso](#)
- [Servizi e integrazioni CloudTrail supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)
- [Ricezione di file di log CloudTrail da più regioni](#) e [Ricezione di file di log CloudTrail da più account](#)

Tutte le operazioni Amazon ElastiCache vengono registrate da CloudTrail e sono documentate nella [Documentazione di riferimento delle API di ElastiCache](#). Ad esempio, le chiamate alle operazioni `CreateCacheCluster`, `DescribeCacheCluster` e `ModifyCacheCluster` generano voci nei file di log di CloudTrail.

Ogni evento o voce di 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 dell'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 `userIdentity` di CloudTrail](#).

Informazioni sulle voci dei file di log di Amazon ElastiCache

Un percorso è una configurazione che consente la distribuzione di eventi come i file di log in un bucket Amazon S3 specificato. I file di log di CloudTrail possono contenere una o più voci di log. Un evento rappresenta una singola richiesta da un'fonte e include informazioni sul operazione richiesta, data e ora dell'operazione, parametri richiesti e così via. I file di log di CloudTrail 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 log di CloudTrail che illustra l'operazione `CreateCacheCluster`.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EXAMPLEEXAMPLEEXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/elasticache-allow",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "elasticache-allow"
  },
  "eventTime": "2014-12-01T22:00:35Z",
  "eventSource": "elasticache.amazonaws.com",
  "eventName": "CreateCacheCluster",
```

```
"awsRegion":"us-west-2",
"sourceIPAddress":"192.0.2.01",
"userAgent":"AWS CLI/ElastiCache 1.10 API 2014-12-01",
"requestParameters":{"
  "numCacheNodes":2,
  "cacheClusterId":"test-memcached",
  "engine":"memcached",
  "aZMode":"cross-az",
  "cacheNodeType":"cache.m1.small",
},
"responseElements":{"
  "engine":"memcached",
  "clientDownloadLandingPage":"https://console.aws.amazon.com/elasticache/
home#client-download:",
  "cacheParameterGroup":{"
    "cacheParameterGroupName":"default.memcached1.4",
    "cacheNodeIdsToReboot":{"
    },
    "parameterApplyStatus":"in-sync"
  },
  "preferredAvailabilityZone":"Multiple",
  "numCacheNodes":2,
  "cacheNodeType":"cache.m1.small",

  "cacheClusterStatus":"creating",
  "autoMinorVersionUpgrade":true,
  "preferredMaintenanceWindow":"thu:05:00-thu:06:00",
  "cacheClusterId":"test-memcached",
  "engineVersion":"1.4.14",
  "cacheSecurityGroups":[
    {
      "status":"active",
      "cacheSecurityGroupName":"default"
    }
  ],
  "pendingModifiedValues":{"
  }
},
"requestID":"104f30b3-3548-11e4-b7b8-6d79ffe84edd",
"eventID":"92762127-7a68-42ce-8787-927d2174cde1"
}
```

L'esempio seguente mostra una voce di log di CloudTrail che illustra l'operazione `DescribeCacheCluster`. Tieni presente che per tutte le chiamate "Describe" Amazon ElastiCache (Describe*), la sezione `ResponseElements` viene rimossa e appare come `null`.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EXAMPLEEXAMPLEEXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/elasticache-allow",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "elasticache-allow"
  },
  "eventTime": "2014-12-01T22:01:00Z",
  "eventSource": "elasticache.amazonaws.com",
  "eventName": "DescribeCacheClusters",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.01",
  "userAgent": "AWS CLI/ElastiCache 1.10 API 2014-12-01",
  "requestParameters": {
    "showCacheNodeInfo": false,
    "maxRecords": 100
  },
  "responseElements": null,
  "requestID": "1f0b5031-3548-11e4-9376-c1d979ba565a",
  "eventID": "a58572a8-e81b-4100-8e00-1797ed19d172"
}
```

L'esempio seguente mostra una voce di log di CloudTrail che registra un'operazione `ModifyCacheCluster`.

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EXAMPLEEXAMPLEEXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/elasticache-allow",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "elasticache-allow"
  },
  "responseElements": null,
  "requestID": "1f0b5031-3548-11e4-9376-c1d979ba565a",
  "eventID": "a58572a8-e81b-4100-8e00-1797ed19d172"
}
```



```
"eventTime":"2014-12-01T22:32:21Z",
"eventSource":"elasticache.amazonaws.com",
"eventName":"ModifyCacheCluster",
"awsRegion":"us-west-2",
"sourceIPAddress":"192.0.2.01",
"userAgent":"AWS CLI/ElastiCache 1.10 API 2014-12-01",
"requestParameters":{"
  "applyImmediately":true,
  "numCacheNodes":3,
  "cacheClusterId":"test-memcached"
},
"responseElements":{"
  "engine":"memcached",
  "clientDownloadLandingPage":"https://console.aws.amazon.com/elasticache/
home#client-download:",
  "cacheParameterGroup":{"
    "cacheParameterGroupName":"default.memcached1.4",
    "cacheNodeIdsToReboot":{"
    },
    "parameterApplyStatus":"in-sync"
  },
  "cacheClusterCreateTime":"Dec 1, 2014 10:16:06 PM",
  "preferredAvailabilityZone":"Multiple",
  "numCacheNodes":2,
  "cacheNodeType":"cache.m1.small",
  "cacheClusterStatus":"modifying",
  "autoMinorVersionUpgrade":true,
  "preferredMaintenanceWindow":"thu:05:00-thu:06:00",
  "cacheClusterId":"test-memcached",
  "engineVersion":"1.4.14",
  "cacheSecurityGroups":[
    {
      "status":"active",
      "cacheSecurityGroupName":"default"
    }
  ],
  "configurationEndpoint":{"
    "address":"test-memcached.example.cfg.use1prod.cache.amazonaws.com",
    "port":11211
  },
  "pendingModifiedValues":{"
    "numCacheNodes":3
  }
},
```

```
"requestID": "807f4bc3-354c-11e4-9376-c1d979ba565a",  
"eventID": "e9163565-376f-4223-96e9-9f50528da645"  
}
```

Quote ElastiCache

L'account AWS dispone delle seguenti quote predefinite, precedentemente definite limiti, per ogni servizio AWS. Salvo dove diversamente specificato, ogni quota si applica a una regione specifica. Se per alcune quote è possibile richiedere aumenti, altre quote non possono essere modificate.

Per visualizzare le quote per ElastiCache, aprire il [Console Service Quotas](#). Nel pannello di navigazione, scegliere Servizi AWS e scegliere ElastiCache.

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 account AWS riporta le quote dell'account in relazione a ElastiCache.

Risorsa	Di default
Cache serverless per regione	40
Snapshot serverless al giorno per cache	24
Nodi per regione	300
Nodi per cluster per tipo di istanza (modalità cluster Redis abilitata)	90
Nodi per partizione (modalità cluster Redis disabilitata)	6
Gruppi di parametri per regione	300
Gruppi di sicurezza per regione	50
Gruppi di sottoreti per regione	300
Sottoreti per gruppo di sottoreti	20
Utenti per gruppo di utenti	100
Numero massimo di utenti	1000

Risorsa	Di default
Numero massimo di gruppi di utenti	100

Riferimento

Gli argomenti presenti in questa sezione illustrano l'utilizzo dell'API Amazon ElastiCache e della sezione ElastiCache del AWS CLI. Sono inclusi anche i messaggi di errore e le notifiche di servizio più frequenti.

- [Utilizzo dell'API ElastiCache](#)
- [Informazioni di riferimento ElastiCache](#)
- [Sezione ElastiCache delAWS CLI Riferimento](#)
- [Messaggi di errore Amazon ElastiCache](#)
- [Notifiche](#)

Utilizzo dell'API ElastiCache

Questa sezione fornisce descrizioni delle attività per l'utilizzo e l'implementazione delle operazioni ElastiCache. Per una descrizione completa di queste operazioni, consulta la [Documentazione di riferimento delle API di Amazon ElastiCache](#)

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 esadecimali (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://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheClusters  
&CacheClusterIdentifier=myCacheCluster  
&SignatureMethod=HmacSHA256  
&SignatureVersion=4  
&Version=2014-12-01
```

Per la stringa di query precedente, è necessario calcolare la firma HMAC sulla seguente stringa.

```
GET\n  
elasticache.amazonaws.com\n  
Action=DescribeCacheClusters  
&CacheClusterIdentifier=myCacheCluster  
&SignatureMethod=HmacSHA256  
&SignatureVersion=4  
&Version=2014-12-01  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Credential=AKIADQKE4SARGYLE%2F20140523%2Fus-west-2%2Felasticache  
%2Faws4_request  
&X-Amz-Date=20141201T223649Z  
&X-Amz-SignedHeaders=content-type%3Bhost%3Buser-agent%3Bx-amz-content-sha256%3Bx-  
amz-date  
content-type:  
host:elasticache.us-west-2.amazonaws.com  
user-agent:CacheServicesAPICommand_Client  
x-amz-content-sha256:  
x-amz-date:
```

Il risultato è la richiesta firmata seguente.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheClusters  
&CacheClusterIdentifier=myCacheCluster  
&SignatureMethod=HmacSHA256  
&SignatureVersion=4  
&Version=2014-12-01  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Credential=AKIADQKE4SARGYLE/20141201/us-west-2/elasticache/aws4_request  
&X-Amz-Date=20141201T223649Z  
&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 relativi argomenti secondari.

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 altri linguaggi, consulta [Codici di esempio e librerie](#).

Risoluzione dei problemi delle applicazioni

ElastiCache fornisce errori specifici e descrittivi per aiutarti a risolvere i problemi mentre interagisci con l'API ElastiCache.

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 nodo `ERROR` nella risposta dall'API ElastiCache.

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: ",
$xml->findvalue("//Error[1]/Code"), "\n", " ",
$xml->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 ElastiCache.

- Verificare che ElastiCache sia eseguito correttamente.

A tale scopo, apri una finestra del browser e invia una richiesta di query al servizio ElastiCache, (ad esempio <https://elasticache.amazonaws.com>). Un'eccezione `MissingAuthenticationTokenException` o un errore server interno 500 conferma che il servizio è disponibile e risponde alle richieste.

- Verificare la struttura della richiesta.

Ogni operazione ElastiCache ha una pagina di riferimento nella documentazione di riferimento dell'API ElastiCache. 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.

Per ElastiCache è disponibile un forum di discussione in cui è possibile cercare soluzioni ai problemi che altri utenti hanno riscontrato lungo il percorso. Per visualizzare il forum, accedi all'indirizzo

<https://forums.aws.amazon.com/>

Impostazione dell'interfaccia a riga di comando di ElastiCache

Questa sezione descrive i prerequisiti per ottenere gli strumenti a riga di comando, dove trovarli, come impostare gli strumenti e il loro ambiente; include una serie di esempi comuni di utilizzo degli strumenti.

Le istruzioni indicate in questo argomento vanno seguite solo se si intende utilizzare la AWS CLI di ElastiCache.

Important

L-interfaccia a riga di comando (CLI) di Amazon ElastiCache non supporta alcun miglioramento ElastiCache dopo la versione API 2014-09-30. Per utilizzare la funzionalità di ElastiCache più recente dalla riga di comando, utilizza il comando [Interfaccia a riga di comando di AWS](#).

Argomenti

- [Prerequisiti](#)
- [Ottenere gli strumenti a riga di comando](#)
- [Configurazione degli strumenti](#)
- [Fornire le credenziali per gli strumenti](#)
- [Variabili di ambiente](#)

Prerequisiti

In questo documento si presuppone che si possa lavorare in un ambiente Linux/UNIX o Windows. Gli strumenti a riga di comando di Amazon ElastiCache funzionano anche su Mac OS X, un ambiente basato su UNIX, ma le istruzioni specifiche per Mac OS X non sono incluse in questa guida.

Per convenzione, tutti i testi a riga di comando hanno un prefisso costituito da un prompt generico **PROMPT>** della riga di comando. Il prompt della riga di comando sul tuo computer potrebbe essere diverso. Utilizziamo anche **\$** per indicare un comando specifico Linux/UNIX e **C:\>** per un comando specifico Windows. L'output di esempio derivante dal comando viene visualizzato subito dopo, senza alcun prefisso.

Java Runtime Environment

Per l'esecuzione, gli strumenti a riga di comando utilizzati in questa guida richiedono almeno la versione 5 di Java. È accettabile l'installazione di JRE o di JDK. Per visualizzare e scaricare i JRE per una serie di piattaforme, tra cui Linux/UNIX e Windows, consulta [Java SE Download](#).

Impostazione della variabile Java Home

Gli strumenti a riga di comando dipendono da una variabile di ambiente (JAVA_HOME) per l'individuazione di Java Runtime. Tale variabile deve essere impostata sul percorso completo della

directory contenente una sottodirectory denominata `bin`, che a sua volta contiene il file eseguibile `java` (su Linux e UNIX) o il file eseguibile `java.exe` (su Windows).

Per impostare la variabile Java Home

1. Impostare la variabile Java Home.

- In Linux e UNIX, immettere il seguente comando:

```
$ export JAVA_HOME=<PATH>
```

- In Windows immetti il seguente comando:

```
C:\> set JAVA_HOME=<PATH>
```

2. Confermare l'impostazione del percorso eseguendo il comando `$JAVA_HOME/bin/java -version` e verificando l'output.

- Su Linux/UNIX, l'aspetto dell'output sarà simile al seguente:

```
$ $JAVA_HOME/bin/java -version
java version "1.6.0_23"
Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
```

- Su Windows, l'aspetto dell'output sarà simile al seguente:

```
C:\> %JAVA_HOME%\bin\java -version
java version "1.6.0_23"
Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
```

Ottenere gli strumenti a riga di comando

Gli strumenti a riga di comando sono disponibili come file `.zip` sul [Sito Web degli strumenti per sviluppatori di ElastiCache](#). Tali strumenti sono scritti in Java e includono script di shell per Windows

2000/XP/Vista/Windows 7, Linux/UNIX e Mac OSX. Il file .zip è autonomo e non occorre installarlo; basta scaricarlo e decomprimerlo in una directory sul computer locale.

Configurazione degli strumenti

Gli strumenti a riga di comando dipendono da una variabile di ambiente (`AWS_ELASTICACHE_HOME`) per l'individuazione delle librerie di support. Prima di poter utilizzare gli strumenti, occorre impostare questa variabile. Impostala sul percorso della directory in cui hai decompresso gli strumenti a riga di comando. La directory è denominata `ElastiCacheCli-A.B.nnnn` (A e B sono i numeri della versione, n è il numero della release) e contiene sottodirectory denominate `bin` e `lib`.

Per impostare la variabile di ambiente `AWS_ELASTICACHE_HOME`

- Aprire una finestra a riga di comando e immettere uno dei seguenti comandi per impostare la variabile di ambiente `AWS_ELASTICACHE_HOME`.
- In Linux e UNIX, immettere il seguente comando:

```
$ export &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

- In Windows immetti il seguente comando:

```
C:\> set &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

Per semplificare l'utilizzo degli strumenti, consigliamo di aggiungere la directory `BIN` degli strumenti al `PATH` di sistema. Il resto della guida presuppone che la directory `BIN` si trovi nel percorso di sistema.

Per aggiungere la directory `BIN` degli strumenti al percorso di sistema

- Immettere i seguenti comandi per aggiungere la directory `BIN` degli strumenti al `PATH` di sistema.
- In Linux e UNIX, immettere il seguente comando:

```
$ export PATH=$PATH:$&AWS;_ELASTICACHE_HOME/bin
```

- In Windows immetti il seguente comando:

```
C:\> set PATH=%PATH%;%&AWS;_ELASTICACHE_HOME%\bin
```

Note

Quando si chiude la finestra dei comandi, le variabili di ambiente Windows vengono resettate. Si possono anche impostare in modo definitivo. Per ulteriori informazioni, consulta la documentazione relativa alla versione di Windows in uso.

Note

I percorsi che contengono uno spazio devono essere racchiusi tra virgolette, ad esempio: "C:\Program Files\Java"

Fornire le credenziali per gli strumenti

Per utilizzare gli strumenti a riga di comando occorrono la chiave di accesso AWS e la chiave di accesso segreta fornite con l'account AWS. Puoi ottenerle tramite la riga di comando o da un file di credenziali presente nel sistema locale.

L'implementazione include un file di modello `${AWS_ELASTICACHE_HOME}/credential-file-path.template` che va modificato con le informazioni. Di seguito i contenuti del file di modello:

```
AWSAccessKeyId=<Write your AWS access ID>  
AWSSecretKey=<Write your AWS secret key>
```

Important

In UNIX, limita le autorizzazioni al proprietario del file delle credenziali:

```
$ chmod 600 <the file created above>
```

Con il file di credenziali configurato, è necessario impostare la variabile di ambiente `AWS_CREDENTIAL_FILE` in modo che gli strumenti di ElastiCache possano trovare le informazioni.

Per impostare la variabile di ambiente `AWS_CREDENTIAL_FILE`

1. Imposta la variabile di ambiente :

- Su Linux e UNIX, impostare la variabile di ambiente utilizzando il comando seguente:

```
$ export &AWS;_CREDENTIAL_FILE=<the file created above>
```

- Su Windows, impostare la variabile di ambiente utilizzando il comando seguente:

```
C:\> set &AWS;_CREDENTIAL_FILE=<the file created above>
```

2. Verificare il corretto funzionamento della configurazione tramite il comando seguente:

```
elasticache --help
```

Dovrebbe comparire la pagina di utilizzo di tutti i comandi ElastiCache.

Variabili di ambiente

Le variabili di ambiente possono essere utili per lo script e la configurazione delle impostazioni predefinite o per la loro provvisoria sostituzione.

Oltre alla variabile di ambiente `AWS_CREDENTIAL_FILE`, la maggior parte degli strumenti API inclusi con l'interfaccia a riga di comando di ElastiCache supporta le seguenti variabili:

- `EC2_REGIONE` la regione AWS da usare.
- `AWS_ELASTICACHE_URL`— L'URL da utilizzare per la chiamata di servizio. Non occorre specificare un endpoint regionale diverso se viene specificata la variabile `EC2_REGION` o se viene trasferito il parametro `--region`.

I seguenti esempi mostrano come impostare la variabile di ambiente `EC2_REGION` per configurare la regione utilizzata dagli strumenti dell'API:

Linux, OS X o Unix

```
$ export EC2_REGION=us-west-1
```

Windows

```
$ set EC2_REGION=us-west-1
```

Messaggi di errore Amazon ElastiCache

Amazon ElastiCache restituisce i messaggi di errore riportati di seguito. Potresti ricevere anche altri messaggi di errore, generati da ElastiCache, diversi servizi AWS o Redis. Per la descrizione dei messaggi di errore non restituiti da ElastiCache, consulta la documentazione originale dei servizi che li hanno generati.

- [Cluster node quota exceeded](#)
- [Customer's node quota exceeded](#)
- [Manual snapshot quota exceeded](#)
- [Insufficient cache cluster capacity](#)

Messaggio di errore: Quota dei nodi di cluster superata. Ogni cluster può annoverare al massimo %n nodi in questa regione."

Causa: hai tentato di creare o modificare un cluster, che, di conseguenza, potrebbe ritrovarsi più di %n nodi.

Soluzione: modifica la richiesta affinché il cluster non annoveri più di %n nodi. Se ti occorrono più di %n nodi, effettua una richiesta apposita tramite il [Modulo di richiesta per nodi di Amazon ElastiCache](#).

Per ulteriori informazioni, consulta [Limiti di Amazon ElastiCache](#) nella Riferimenti generali di Amazon Web Services.

Messaggi di errore: Quota di nodi del cliente superata. Puoi impostare al massimo %n nodi in questa regione . In alternativa, "Hai già raggiunto la quota massima di %s nodi in questa regione".

Causa: in seguito al tentativo di creazione o modifica di un cluster, l'account potrebbe annoverare più di %n nodi tra i cluster della regione.

Soluzione: modifica la richiesta per non superare la quota totale di %n nodi ripartiti tra i cluster della regione dell'account. Se ti occorrono più di %n nodi, effettua una richiesta apposita tramite il [Modulo di richiesta per nodi di Amazon ElastiCache](#).

Per ulteriori informazioni, consulta [Limiti di Amazon ElastiCache](#) nella Riferimenti generali di Amazon Web Services.

Messaggi di errore: "È stato raggiunto il numero massimo consentito di snapshot manuali del cluster nelle 24 ore" " oppure "Il numero di snapshot manuali di questo nodo nelle 24 ore ha raggiunto il limite massimo di %n"

Causa: hai tentato di acquisire una snapshot manuale di un cluster pur avendo già superato il numero massimo di snapshot manuali consentite nell'arco di 24 ore.

Soluzione: attendi 24 ore, prima di ritentare l'acquisizione di una nuova snapshot manuale del cluster. In caso di immediata necessità, puoi acquisire la snapshot di un altro nodo di cluster con gli stessi dati.

Messaggio di errore: InsufficientCacheClusterCapacity

Causa: AWS al momento non dispone di sufficiente capacità on demand per soddisfare la richiesta.

Soluzione::

- Attendi alcuni minuti, quindi invia di nuovo la richiesta; la capacità può variare frequentemente.
- Inviare una nuova richiesta con un numero ridotto di nodi o shard (gruppi di nodi). Ad esempio, se si effettua un'unica richiesta di avvio di 15 nodi, tentare creando tre richieste di cinque nodi oppure 15 richieste di un nodo.
- Se si sta avviando un cluster, inviare una nuova richiesta senza specificare alcuna zona di disponibilità.

- Se si sta avviando un cluster, inviare una nuova richiesta utilizzando un tipo di nodo diverso (che sarà possibile ridimensionare in un secondo momento). Per ulteriori informazioni, consulta [Scalabilità per Redis ElastiCache](#).

Notifiche

In questo argomento vengono descritte le notifiche ElastiCache che potrebbero essere di tuo interesse. Una notifica è una situazione o un evento nella maggior parte dei casi temporaneo, che dura finché non viene individuata o implementata una soluzione. Le notifiche hanno in genere una data di inizio e una data di risoluzione, dopo la quale la notifica non è più rilevante. Una notifica potrebbe o meno essere rilevante per te. Ti consigliamo una linea guida di implementazione che, se seguita, migliora le prestazioni del cluster.

Le notifiche non annunciano caratteristiche o funzionalità nuove o migliorate di ElastiCache.

Notifiche generali ElastiCache

Attualmente non sono disponibili notifiche ElastiCache in sospenso non specifiche del motore.

ElastiCache per notifiche specifiche Redis

Attualmente, non ci sono notifiche ElastiCache for Redis in sospenso.

ElastiCache per la cronologia della documentazione Redis

- Versione API: 02-02-2015
- Ultimo aggiornamento della documentazione: 27 novembre 2023

La tabella seguente descrive le modifiche importanti apportate in ogni versione della per Redis dopo marzo 2018. Per ricevere notifiche sugli aggiornamenti della documentazione, puoi sottoscrivere il feed RSS.

Aggiornamenti recenti ElastiCache per Redis

Modifica	Descrizione	Data
ElastiCache per Redis ha aggiunto il supporto per nodi C7gn di dimensioni aggiuntive	ElastiCache per Redis ha aggiunto il supporto per nodi C7gn di dimensioni aggiuntive.	10 gennaio 2024
ElastiCache for Redis ora supporta la creazione di cache serverless	Ora puoi creare le cache serverless che semplificano la gestione delle cache e si dimensionano istantaneamente per supportare le applicazioni più esigenti. Per ulteriori informazioni, consulta Scelta tra le opzioni di implementazione . Come parte di questa funzionalità, sono state aggiunte nuove autorizzazioni a <code>ElastiCacheServiceRolePolicy</code> e <code>AmazonElastiCacheFullAccess</code> per consentire l'associazione di cache serverless con endpoint VPC gestiti. Inoltre, sono state aggiunte le autorizzazioni per	27 novembre 2023

supportare un'esperienza di console modificata utilizzando la policy AmazonElastiCacheFullAccess .

[ElastiCache for Redis ora supporta la modifica della modalità cluster](#)

Ora è possibile migrare i cluster da Modalità cluster disabilitata (CMD) a Modalità cluster abilitata (CME). Per ulteriori informazioni, consultare [Modifica della modalità cluster](#).

11 maggio 2023

[ElastiCache for Redis ora supporta la modifica delle impostazioni di crittografia in transito](#)

Ora è possibile modificare la configurazione TLS dei cluster Redis senza la necessità dover ricreare o effettuare nuovamente il provisioning dei cluster o influire sulla disponibilità delle applicazioni. Per ulteriori informazioni, consulta [Abilitazione della crittografia dei dati in transito per un cluster esistente](#).

28 dicembre 2022

[ElastiCache for Redis ora supporta l'autenticazione degli utenti tramite IAM](#)

L'autenticazione IAM consente di autenticare una connessione a ElastiCache Redis utilizzando identità IAM. AWS Ciò consente di consolidare il modello di sicurezza e semplificare molte attività di sicurezza amministrative. Per ulteriori informazioni, consulta [Autenticazione con IAM](#).

16 novembre 2022

[ElastiCache per Redis ora supporta Redis 7](#)

Questa versione offre diverse nuove funzionalità ad Amazon ElastiCache for Redis: funzioni Redis, miglioramenti ACL e Sharded Pub/Sub. [Per ulteriori informazioni, consulta la versione 7.0 di Redis. ElastiCache](#)

8 novembre 2022

[ElastiCache per Redis ora supporta IPV6](#)

ElastiCache supporta le versioni 4 e 6 del protocollo Internet (IPv4 e IPv6), che consente di configurare il cluster in modo che accetti solo connessioni IPv4, solo connessioni IPv6 o entrambe le connessioni IPv4 e IPv6 (dual-stack). IPv6 è supportato per i carichi di lavoro che utilizzano il motore Redis dalla versione 6.2 in avanti su tutte le istanze sviluppate e sul [sistema Nitro](#). Non sono previsti costi aggiuntivi per l'accesso tramite IPv6. ElastiCache Per ulteriori informazioni, consulta [Scelta di un tipo di rete](#).

7 novembre 2022

[ElastiCache per Redis ora supporta il formato nativo di JavaScript Object Notation \(JSON\)](#)

Il formato nativo di JavaScript Object Notation (JSON) è un modo semplice e senza schemi per codificare set di dati complessi all'interno di cluster Redis. È possibile archiviare e accedere in modo nativo ai dati utilizzando il formato JavaScript Object Notation (JSON) all'interno dei cluster Redis e aggiornare i dati JSON archiviati in tali cluster, senza dover gestire codice personalizzato per serializzarli e deserializzarli. Per ulteriori informazioni, consulta [Nozioni di base su JSON](#).

25 maggio 2022

[ElastiCache ora supporta PrivateLink](#)

AWS PrivateLink consente di accedere in modo privato alle operazioni ElastiCache API senza un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione AWS Direct Connect. Per ulteriori informazioni, consulta [Amazon ElastiCache API e interfaccia VPC endpoints \(AWS PrivateLink\)](#) per Redis o [ElastiCache Amazon API e interfaccia VPC endpoints \(\)](#) per Memcached. AWS PrivateLink

24 gennaio 2022

[ElastiCache per Redis ora supporta Redis 6.2 e Data Tiering](#)

Amazon ElastiCache for Redis 23 novembre 2021 introduce la nuova versione del motore Redis supportato da Amazon. ElastiCache ElastiCache per Redis 6.2 include miglioramenti delle prestazioni per i cluster abilitati a TLS che utilizzano tipi di nodi x86 con 8 o più vCPU o tipi di nodi Graviton2 con 4 o più vCPU. ElastiCache for Redis introduce anche il tiering dei dati. 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 ElastiCache la versione 6.2 \(avanzata\) di Redis e Data tiering.](#)

[Support per Auto Scaling](#)

ElastiCache for Redis ora supporta Auto Scaling. ElastiCache per Redis la scalabilità automatica è la capacità di aumentare o diminuire automaticamente gli shard o le repliche desiderate nel servizio ElastiCache for Redis. ElastiCache sfrutta il servizio Application Auto Scaling per fornire questa funzionalità. Per ulteriori informazioni, consulta [Auto Scaling ElastiCache for Redis clusters](#).

19 agosto 2021

[Support per la consegna dei registri Redis Slow](#)

ElastiCache ora consente di eseguire lo streaming di Redis SLOWLOG su una delle due destinazioni: Amazon Data Firehose o Amazon Logs. CloudWatch Per ulteriori informazioni, consulta [Consegna dei registri](#).

22 Aprile 2021

[Support per assegnazione di tag a risorse e chiavi di condizione](#)

ElastiCache ora supporta l'etichettatura per aiutarti a gestire i cluster e altre risorse. ElastiCache Per ulteriori informazioni, consulta [Taggare](#) le risorse. ElastiCache introduce anche il supporto per le chiavi di condizione. Puoi specificare le condizioni che determinano il modo in cui una policy IAM viene applicata. Per ulteriori informazioni, consulta [Utilizzo delle chiavi di condizione](#).

7 Aprile 2021

[ElastiCache è ora disponibile su AWS Outposts](#)

[AWS Outposts offre](#) AWS servizi, infrastrutture e modelli operativi nativi praticamente a qualsiasi data center, spazio di co-location o struttura locale. Puoi eseguire la distribuzione ElastiCache su Outposts per configurare, gestire e utilizzare la cache in locale, proprio come faresti nel cloud. Per ulteriori informazioni, consulta [Utilizzo di Outposts](#) per Redis o [Utilizzo di Outposts](#) per Memcached.

8 ottobre 2020

[ElastiCache ora supporta Redis 6](#)

Amazon ElastiCache for Redis introduce la nuova versione del motore Redis supportato da Amazon. ElastiCache Questa versione include [Autenticazione degli utenti con il controllo accessi basato sui ruoli](#), support senza versione, caching lato client e significativi miglioramenti operativi. Per ulteriori informazioni, consulta la [versione 6.0 \(ElastiCache Enhanced\) di Redis](#).

7 ottobre 2020

[ElastiCache ora supporta Local Zones](#)

Una zona locale è un'estensione di una AWS regione geograficamente vicina agli utenti. È possibile estendere qualsiasi cloud privato virtuale (VPC) da una AWS regione principale alle Local Zones creando una nuova sottorete e assegnandola a una Local Zone. Per ulteriori informazioni, consulta [Utilizzo di Local Zones](#).

25 settembre 2020

[ElastiCache for Redis ora supporta la scalabilità dell'ambiente Redis Cluster fino a 500 nodi o 500 shard](#)

La modalità Redis Cluster rende possibili configurazioni che è possibile utilizzare per partizionare i dati tra più partizioni e offre una migliore scalabilità, prestazioni e disponibilità. Questa funzionalità è disponibile su Amazon ElastiCache for Redis dalla versione 5.0.6 in poi in tutte le AWS regioni e per tutti gli ambienti Redis Cluster esistenti e nuovi ElastiCache. Per ulteriori informazioni, consulta [Nodi e partizioni Redis](#).

13 agosto 2020

[ElastiCache ora supporta le autorizzazioni a livello di risorsa](#)

Ora puoi limitare l'ambito delle autorizzazioni di un utente specificando le ElastiCache risorse in una policy (IAM). AWS Identity and Access Management Per ulteriori informazioni, consulta [Autorizzazioni a livello di risorsa](#).

12 agosto 2020

[ElastiCache per Redis aggiunge parametri Amazon CloudWatch aggiuntivi](#)

ElastiCache for Redis ora supporta nuove CloudWatch metriche, tra cui e. PubSubCmds HyperLogLogBasedCmds Per un elenco completo, vedere [Metriche per Redis](#).

10 giugno 2020

[ElastiCache ora supporta l'aggiornamento automatico dei cluster ElastiCache](#)

Amazon ElastiCache ora supporta l'aggiornamento automatico dei ElastiCache cluster dopo la «data di scadenza consigliata» dell'aggiornamento del servizio. ElastiCache utilizza la finestra di manutenzione per pianificare l'aggiornamento automatico dei cluster applicabili. Per ulteriori informazioni, vedere [Aggiornamenti self-service](#).

13 maggio 2020

[ElastiCache for Redis ora supporta Global Datastore for Redis](#)

La funzionalità Global Datastore for Redis offre una replica completamente gestita, veloce, affidabile e sicura in tutte le regioni. AWS Utilizzando questa funzionalità, puoi creare cluster di replica di lettura interregionali per Redis ElastiCache per consentire letture a bassa latenza e disaster recovery tra le regioni. AWS È possibile creare, modificare e descrivere un datastore globale. Puoi anche aggiungere o rimuovere AWS regioni dal tuo datastore globale e promuovere una regione come principale all'interno di un datastore globale. AWS Per ulteriori informazioni, consulta [Replica tra AWS regioni utilizzando Global Datastore](#).

16 marzo 2020

[ElastiCache per Redis ora supporta la versione Redis 5.0.6](#)

Per ulteriori informazioni, consulta la [versione 5.0.6 ElastiCache di Redis \(Enhanced\)](#).

18 dicembre 2019

[Amazon ElastiCache ora supporta i nodi di cache T3 Standard](#)

Ora puoi lanciare i nodi di cache T3 Standard espandibili per uso generico di nuova generazione in Amazon ElastiCache. Le istanze T3-Standard di Amazon EC2 offrono un livello base di prestazioni della CPU con la possibilità di incrementarne l'utilizzo quando necessari o fino all'esaurimento dei crediti accumulati. Per ulteriori informazioni, consulta [Tipi di nodi supportati](#).

12 novembre 2019

[Amazon ElastiCache ora supporta la modifica del token AUTH su un server esistente ElastiCache per Redis](#)

ElastiCache per Redis 5.0.6 ora ti consente di modificare e i token di autenticazione impostando e ruotando nuovi token. È ora possibile modificare i token attivi quando sono in uso. È possibile anche aggiungere nuovi token ai cluster esistenti abilitati con la crittografia in transito precedentemente configurati senza token di autenticazione. Questo è un processo a due fasi con cui è possibile impostare e ruotare il token senza interrompere le richieste client. Questa funzionalità non è attualmente supportata su. AWS CloudFormation Per ulteriori informazioni, consulta [Autenticazione degli utenti con il comando AUTH Redis](#).

30 ottobre 2019

[Amazon ElastiCache ora supporta la migrazione dei dati online da Redis su Amazon EC2](#)

Ora puoi utilizzare Online Migration per migrare i tuoi dati da Redis self-hosted su Amazon EC2 ad Amazon. ElastiCache [Per ulteriori informazioni, consulta Online Migration to. ElastiCache](#)

28 ottobre 2019

[ElastiCache for Redis introduce la scalabilità verticale online per la modalità Redis Cluster.](#)

Ora puoi scalare verso l'alto o verso il basso il tuo Redis Cluster condiviso su richiesta. ElastiCache for Redis ridimensiona il cluster modificando il tipo di nodo, mentre il cluster continua a rimanere online e a soddisfare le richieste in arrivo. Per ulteriori informazioni, consulta [Online Vertical Scaling by Modifying Node Type](#).

20 agosto 2019

[ElastiCache for Redis ora consente agli utenti di utilizzare un endpoint di lettura singolo per il cluster Amazon ElastiCache for Redis.](#)

Questa funzionalità consente di indirizzare tutto il traffico di lettura verso il cluster ElastiCache for Redis tramite un unico endpoint a livello di cluster per sfruttare il bilanciamento del carico e una maggiore disponibilità. Per maggiori informazioni, consulta [Trovare gli endpoint di connessione](#).

13 giugno 2019

[ElastiCache for Redis ora consente agli utenti di applicare gli aggiornamenti del servizio secondo la propria pianificazione](#)

Con questa funzionalità, puoi scegliere di applicare gli aggiornamenti del servizio disponibili al momento desiderato e non solo durante le finestre di manutenzione. Ciò ridurrà al minimo le interruzioni del servizio, in particolare durante i flussi aziendali di picco, e contribuirà a garantire la conformità se il cluster utilizza programmi di conformità supportati. ElastiCache Per ulteriori informazioni, consulta [Aggiornamenti self-service in Amazon ElastiCache](#) e [Convalida della conformità per Amazon](#). ElastiCache

4 giugno 2019

[ElastiCache Offerte di istanze riservate standard: pagamento anticipato parziale, pagamento anticipato completo e nessun pagamento anticipato.](#)

Le istanze riservate offrono la flessibilità necessaria per prenotare un' ElastiCache istanza Amazon per un periodo di uno o tre anni in base al tipo di istanza e AWS alla regione. Per ulteriori informazioni, consulta [Gestione dei costi con nodi riservati](#).

18 gennaio 2019

[ElastiCache per il supporto Redis per un massimo di 250 nodi per cluster Redis](#)

Il limite di nodi o shard può essere aumentato fino a un massimo di 250 ElastiCache per cluster Redis. Per ulteriori informazioni, consulta [Shard](#).

19 novembre 2018

[ElastiCache per il supporto Redis per l'autofailover e il backup e il ripristino su tutti i nodi T2](#)

ElastiCache for Redis introduce il supporto per l'autofailover, la creazione di istantanee e il backup e il ripristino su tutti i nodi T2. Per ulteriori informazioni, consulta [ElastiCache Backup and Restore and Snapshot di Redis](#).

19 novembre 2018

[ElastiCache per il supporto Redis per i nodi M5 e R5](#)

ElastiCache for Redis ora supporta i nodi M5 e R5, tipi di istanza generici e ottimizzati per la memoria basati sul sistema Nitro. AWS Per ulteriori informazioni, consulta [Tipi di nodi supportati](#).

23 ottobre 2018

[Support per la modifica dinamica del numero di repliche di lettura](#)

ElastiCache for Redis ha aggiunto il supporto per l'aggiunta e la rimozione di repliche di lettura da qualsiasi cluster senza tempi di inattività del cluster. Per ulteriori informazioni su queste e altre modifiche introdotte in questa versione, consulta [Modifica del numero di repliche](#) nella guida ElastiCache per l'utente di Redis. Vedi anche [DecreaseReplicaCount](#) [IncreaseReplicaCount](#) nella ElastiCache API Reference.

17 settembre 2018

Certificazione di conformità FedRAMP	ElastiCache for Redis è ora certificato per la conformità a FedRAMP. Per ulteriori informazioni, consulta Convalida della conformità per Amazon ElastiCache .	30 agosto 2018
Aggiornamenti del motore Redis (modalità cluster abilitata)	Amazon ElastiCache for Redis ha aggiunto il supporto per l'aggiornamento delle versioni del motore Redis (abilitata per la modalità cluster). Per ulteriori informazioni, consulta la sezione dedicata all' aggiornamento delle versioni di un motore .	20 agosto 2018
Certificazione di conformità PCI DSS	ElastiCache for Redis è ora certificato per la conformità a PCI DSS. Per ulteriori informazioni, consulta Convalida della conformità per Amazon ElastiCache .	5 luglio 2018
Support ElastiCache per Redis 4.0.10	ElastiCache for Redis ora supporta Redis 4.0.10, inclusa la crittografia e il ridimensionamento online del cluster in un'unica versione. Per ulteriori informazioni, consulta Redis versione 4.0.10 (ElastiCache Enhanced) .	14 giugno 2018

[Guida per l'utente ristrutturata](#) [La guida ElastiCache utente singola è ora ristrutturata in modo che esistano guide utente separate per Redis \(per Redis User Guide\) e ElastiCache per Memcached \(per Memcached User Guide\).](#) [ElastiCache](#) La struttura della documentazione nella sezione [AWS CLI Command Reference: elasticache](#) e [Amazon ElastiCache API Reference](#) rimane invariata.

20 Aprile 2018

[Support per il parametro EngineCPUUtilization](#)

ElastiCache for Redis ha aggiunto una nuova metrica `EngineCPUUtilization`, che riporta la percentuale della capacità della CPU attualmente utilizzata. Per ulteriori informazioni, consulta la pagina relativa ai [parametri di Redis](#).

9 aprile 2018

La tabella seguente descrive le modifiche importanti alla per Redis prima di marzo 2018.

Modifica	Descrizione	Data della modifica
Support per Regione Asia Pacifico (Osaka-Locale)	ElastiCache ha aggiunto il supporto per la regione Asia Pacifico (locale di Osaka). La regione Asia Pacifico (Osaka) attualmente supporta un'unica zona di disponibilità e funziona solo tramite invito. Per ulteriori informazioni, consulta gli argomenti seguenti: <ul style="list-style-type: none"> Regioni supportate 	12 febbraio 2018

Modifica	Descrizione	Data della modifica
	<ul style="list-style-type: none">• Tipi di nodi di cache supportati	
Support per l'UE (Parigi).	ElastiCache ha aggiunto il supporto per la regione UE (Parigi). Per ulteriori informazioni, consulta gli argomenti seguenti: <ul style="list-style-type: none">• Regioni supportate• Tipi di nodi di cache supportati	18 dicembre 2017
Support per la regione Cina (Ningxia)	Amazon ElastiCache ha aggiunto il supporto per la regione Cina (Ningxia). Per ulteriori informazioni, consulta gli argomenti seguenti: <ul style="list-style-type: none">• Regioni supportate• Tipi di nodi di cache supportati	11 dicembre 2017
Support per i ruoli collegati ai servizi	Questa versione ElastiCache ha aggiunto il supporto per Service Linked Roles (SLR). Per ulteriori informazioni, consulta gli argomenti seguenti: <ul style="list-style-type: none">• Utilizzo di ruoli collegati ai servizi per Amazon ElastiCache• Configura le tue autorizzazioni (solo nuovi ElastiCache utenti)	7 dicembre 2017

Modifica	Descrizione	Data della modifica
Support per i tipi di nodi R4	<p>Questa versione di supporto ElastiCache aggiuntivo o supporta i tipi di nodi R4 in tutte le AWS regioni supportate da ElastiCache. È possibile acquistare i tipi di nodo R4 in modalità on demand o come nodi di cache riservati. Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none">• Tipi di nodi di cache supportati• Parametri specifici del tipo di nodo di Redis	20 novembre 2017
ElastiCache per Redis 3.2.10 e supporto per il resharding online	<p>Amazon ElastiCache for Redis aggiunge il supporto ElastiCache per Redis 3.2.10. ElastiCache for Redis introduce anche il ridimensionamento online del cluster per aggiungere o rimuovere frammenti dal cluster mentre continua a soddisfare le richieste di I/O in entrata. Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none">• Ridimensionamento di cluster online• Reshard online e ribilanciamento della partizione per Redis (modalità cluster abilitata)	9 novembre 2017
Conformità HIPAA	<p>ElastiCache per Redis la versione 3.2.6 è ora certificata per l'idoneità HIPAA quando la crittografia è abilitata sul cluster. Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none">• Convalida della conformità per Amazon ElastiCache• Sicurezza dei dati in Amazon ElastiCache	2 Novembre 2017

Modifica	Descrizione	Data della modifica
ElastiCache per Redis 3.2.6 e supporto per la crittografia	<p>ElastiCache aggiunge il supporto ElastiCache per Redis 3.2.6, che include due funzionalità di crittografia:</p> <ul style="list-style-type: none">• La crittografia in transito crittografa i dati quando sono in transito, ad esempio tra i nodi in un cluster o tra un cluster e l'applicazione.• La crittografia dei dati inattivi esegue la crittografia su disco dei dati durante operazioni di sincronizzazione e backup. <p>Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none">• Sicurezza dei dati in Amazon ElastiCache• ElastiCache supportate per le versioni Redis	25 ottobre 2017
Argomento dei modelli di connessione	<p>ElastiCache la documentazione aggiunge un argomento che tratta vari modelli di accesso a un ElastiCache cluster in un Amazon VPC.</p> <p>Per ulteriori informazioni, consulta Modelli di accesso a una cache ElastiCache in un Amazon VPC nella Guida per l'utente di ElastiCache .</p>	24 Aprile 2017

Modifica	Descrizione	Data della modifica
Support per il test del failover automatico	<p>ElastiCache aggiunge il supporto per testare il failover automatico su cluster Redis che supportano o la replica. Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none">• Test del failover automatico nella Guida per l'utente di ElastiCache .• TestFailover nel documento di riferimento delle API ElastiCache• test-failover nella documentazione di riferimento dell'AWS CLI .	4 Aprile 2017
Ripristino di Redis potenziato	<p>ElastiCache aggiunge funzionalità avanzate di backup e ripristino Redis con ridimensionamento del cluster. Questa funzione consente il ripristino del backup in un cluster con un numero di partizioni diverso rispetto a quello del cluster utilizzato per creare il backup. (Per l'API e la CLI questa caratteristica può ripristinare un numero diverso di gruppi di nodi piuttosto che un numero diverso di partizioni.) Questo aggiornamento supporta anche diverse configurazioni di slot Redis. Per ulteriori informazioni, consulta Ripristino da un backup in una nuova cache.</p>	15 marzo 2017

Modifica	Descrizione	Data della modifica
Nuovo parametro di gestione della memoria Redis	<p>ElastiCache aggiunge un nuovo parametro <code>Redisreserved-memory-percent</code>, che semplifica la gestione della memoria riservata. Questo parametro è disponibile in tutte le versioni di ElastiCache for Redis. Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none">• Gestione della memoria prenotata• Nuovi parametri per Redis 3.2.4	15 marzo 2017
Support per la regione Europa occidentale (Londra)	<p>ElastiCache aggiunge il supporto per la regione UE (Londra). Al momento sono supportati solo i tipi di nodi T2 e M4. Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none">• Regioni supportate• Tipi di nodi di cache supportati	13 dicembre 2016
Support per la regione Canada (Montreal)	<p>ElastiCache aggiunge il supporto per la regione Canada (Montreal). Attualmente in questa regione sono supportati solo i tipi di nodi M4 e T2. AWS Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none">• Regioni supportate• Tipi di nodi di cache supportati	8 dicembre 2016
Support per i tipi di nodi M4 e R3	<p>ElastiCache aggiunge il supporto per i tipi di nodi R3 e M4 nella regione del Sud America (San Paolo) e i tipi di nodi M4 nella regione della Cina (Pechino). Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none">• Regioni supportate• Tipi di nodi di cache supportati	1 Novembre 2016

Modifica	Descrizione	Data della modifica
support per la regione Stati Uniti orientali 2 (Ohio)	<p>ElastiCache aggiunge il supporto per la regione Stati Uniti orientali (Ohio) (us-east-2) con i tipi di nodi M4, T2 e R3. Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none">• Regioni supportate• Tipi di nodi di cache supportati	17 ottobre 2016
Support per il cluster Redis	<p>ElastiCache aggiunge il supporto per Redis Cluster (avanzato). I clienti che utilizzano Redis Cluster possono ripartire i propri dati tra un massimo di 15 partizioni (gruppi di nodi). Ogni partizione supporta la replica (fino a cinque repliche di lettura per partizione). Il failover automatico del cluster Redis impiega circa un quarto del tempo rispetto alle versioni precedenti.</p> <p>Questa versione include una console di gestione riprogettata, che vanta una terminologia in linea con gli standard di settore.</p> <p>Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none">• Confronto tra Memcached e Redis• ElastiCache per componenti e funzionalità Redis: annotare le sezioni su Nodi, Frammenti, Cluster e Replica.• Terminologia di ElastiCache per Redis	12 Ottobre 2016

Modifica	Descrizione	Data della modifica
Support per i tipi di nodi M4	ElastiCache aggiunge il supporto per la famiglia di tipi di nodi M4 nella maggior parte delle AWS regioni supportate da. ElastiCache È possibile acquistare i tipi di nodo M4 in modalità on demand o come nodi di cache riservati. Per ulteriori informazioni, consulta gli argomenti seguenti: <ul style="list-style-type: none">• Tipi di nodi di cache supportati• Parametri specifici del tipo di nodo di Redis	3 agosto 2016
Support per la regione di Mumbai	ElastiCache aggiunge il supporto per la regione Asia Pacifico (Mumbai). Per ulteriori informazioni, consulta gli argomenti seguenti: <ul style="list-style-type: none">• Tipi di nodi di cache supportati• Parametri specifici del tipo di nodo di Redis	27 giugno 2016
Esportazione di snapshot	ElastiCache aggiunge la possibilità di esportare un'istantanea Redis in modo da potervi accedere dall'esterno. ElastiCache Per ulteriori informazioni, consulta gli argomenti seguenti: <ul style="list-style-type: none">• Esportazione di un backup nella Amazon ElastiCache User Guide• CopySnapshot nell'Amazon ElastiCache API Reference	26 maggio 2016
Dimensionamento verso l'alto del tipo di nodo	ElastiCache aggiunge la possibilità di scalare il tipo di nodo Redis. Per ulteriori informazioni, consulta Scalabilità per Redis ElastiCache .	24 marzo 2016
Aggiornamento del motore semplificato	ElastiCache aggiunge la possibilità di aggiornare facilmente il motore di cache Redis. Per ulteriori informazioni, consulta Versioni e aggiornamenti del motore .	22 marzo 2016

Modifica	Descrizione	Data della modifica
Support per i tipi di nodi R3	ElastiCache aggiunge il supporto per i tipi di nodi R3 nella regione Cina (Pechino) e nella regione del Sud America (San Paolo). Per ulteriori informazioni, consulta Tipi di nodi di cache supportati .	16 marzo 2016
Accesso ElastiCache tramite una funzione Lambda	È stato aggiunto un tutorial sulla configurazione di una funzione Lambda per ElastiCache l'accesso in un Amazon VPC. Per ulteriori informazioni, consulta Esercitazioni e video relativi a ElastiCache .	12 febbraio 2016
Support per Redis 2.8.24	ElastiCache aggiunge il supporto per la versione 2.8.24 di Redis con miglioramenti aggiunti dopo Redis 2.8.23. I miglioramenti includono la correzione di alcuni bug e il supporto per la registrazione di indirizzi di accesso alla memoria con errori. Per ulteriori informazioni, consulta gli argomenti seguenti: <ul style="list-style-type: none">• ElastiCache for Redis versione 2.8.24 (potenziata)• Note di rilascio di Redis 2.8	20 gennaio 2016
Support per la regione Asia Pacifico (Seoul).	ElastiCache aggiunge il supporto per la regione Asia Pacifico (Seoul) (ap-northeast-2) con i tipi di nodi t2, m3 e r3.	6 gennaio 2016
Modifica ElastiCache della console Amazon.	Poiché le versioni più recenti di Redis offrono un'esperienza utente migliore e più stabile, le versioni Redis 2.6.13, 2.8.6 e 2.8.19 non sono più elencate nella Console di gestione. ElastiCache Per altre opzioni e ulteriori informazioni, consulta ElastiCache supportate per le versioni Redis .	15 dicembre 2015

Modifica	Descrizione	Data della modifica
Support per Redis 2.8.23.	ElastiCache aggiunge il supporto per la versione Redis 2.8.23 con miglioramenti aggiunti rispetto a Redis 2.8.22. I miglioramenti includono la correzione di alcuni bug e il supporto per il nuovo parametro <code>close-on-slave-write</code> che, se abilitato, disconnette i client che tentano di scrivere in una replica di sola lettura. Per ulteriori informazioni, consulta ElastiCache for Redis versione 2.8.23 (potenziata) .	13 Novembre 2015

Modifica	Descrizione	Data della modifica
Support per Redis 2.8.22.	<p>ElastiCache aggiunge il supporto per la versione Redis 2.8.22 con miglioramenti e miglioramenti aggiunti rispetto alla versione 2.8.21. ElastiCache Miglioramenti:</p> <ul style="list-style-type: none">• Implementazione di un processo di salvataggio senza fork che consente di salvare quando la scarsa memoria disponibile rischia di bloccare un salvataggio con fork.• SaveInProgressReplicationBytesMetriche aggiuntive e. CloudWatch• Applicazione del parametro Redis <code>rep1-backlog-size</code> a tutti i cluster, per consentire la sincronizzazione parziale. <p>Per l'elenco completo delle modifiche e ulteriori informazioni, consulta ElastiCache for Redis versione 2.8.22 (potenziata).</p> <p>Questa versione della documentazione include una riorganizzazione della documentazione e la rimozione della documentazione dell'interfaccia a riga di ElastiCache comando (CLI). Per l'utilizzo della riga di comando, fare riferimento alla AWS riga di comando per. ElastiCache</p>	28 settembre 2015
Support per Redis 2.8.21	<p>ElastiCache aggiunge il supporto per la versione 2.8.21 di Redis e i miglioramenti di Redis rispetto alla versione 2.8.19. Questa versione di Redis vanta la risoluzione di diversi bug. Per ulteriori informazioni, consulta le note di rilascio di Redis 2.8.</p>	29 luglio 2015

Modifica	Descrizione	Data della modifica
Nuovo argomento: Accesso dall'esterno ElastiCache AWS	Aggiunto un nuovo argomento su come accedere alle ElastiCache risorse dall'esterno AWS. Per ulteriori informazioni, vedere Accesso ElastiCache dall'esterno AWS .	9 luglio 2015
Aggiunta di messaggi relativi alla sostituzione di nodi	ElastiCache aggiunge tre messaggi relativi alla sostituzione pianificata dei nodi, ElastiCache: NodeReplacementScheduled, ElastiCache: NodeReplacementRescheduled, ElastiCache: NodeReplacementCanceled. Per ulteriori informazioni e azioni che puoi intraprendere quando è pianificata la sostituzione di un nodo, ElastiCache consulta Notifiche di eventi Amazon SNS .	11 giugno 2015

Modifica	Descrizione	Data della modifica
Supporto per Redis 2.8.19.	<p>ElastiCache aggiunge il supporto per la versione 2.8.19 di Redis e i miglioramenti di Redis dalla versione 2.8.6. Supporto garantito per:</p> <ul style="list-style-type: none">• La struttura HyperLogLog dei dati, con i comandi Redis PFADD, PFCOUNT e PFMERGE.• Query di intervallo lessicografiche con i nuovi comandi ZRANGEBYLEX, ZLEXCOUNT e ZREMRANGEBYLEX.• Introdotte varie correzioni di bug, grazie alle quali si impedisce che un nodo primario invii dati non aggiornati ai nodi di replica, fallendo la sincronizzazione del primario quando un processo di salvataggio in background (bgsave) figlio termina inaspettatamente. <p>Per ulteriori informazioni su HyperLogLog, consulta la nuova struttura dati di Redis: the. HyperLogLog</p> <p>Per ulteriori informazioni su PFADD, PFCOUNT e PFMERGE, consultate la documentazione Redis e fate clic su. HyperLogLog</p>	11 marzo 2015
Supporto per i tag di allocazione dei costi	<p>ElastiCache aggiunge il supporto per i tag di allocazione dei costi. Per ulteriori informazioni, consulta Monitoraggio dei costi con i tag di allocazione dei costi.</p>	9 febbraio 2015
Supporto per la AWS GovCloud regione (Stati Uniti occidentali)	<p>ElastiCache aggiunge il supporto per la regione AWS GovCloud (Stati Uniti occidentali) (us-gov-west-1).</p>	29 gennaio 2015

Modifica	Descrizione	Data della modifica
Support per la regione Europa (Francoforte)	ElastiCache aggiunge il supporto per la regione Europa (Francoforte) (eu-central-1).	19 gennaio 2015
Support Multi-AZ per gruppi di replica Redis	ElastiCache aggiunge il supporto per Multi-AZ dal nodo primario a una replica di lettura in un gruppo di replica Redis. ElastiCache monitora lo stato del gruppo di replica. In caso di errore della replica principale, promuove ElastiCache automaticamente una replica a principale, quindi sostituisce la replica. Per ulteriori informazioni, consulta Riduzione dei tempi di inattività in ElastiCache for Redis con Multi-AZ .	24 Ottobre 2014
AWS CloudTrail è supportata la registrazione delle chiamate API	ElastiCache aggiunge il supporto per l'utilizzo AWS CloudTrail per registrare tutte le chiamate ElastiCache API. Per ulteriori informazioni, consulta Registrazione delle chiamate all'API Amazon ElastiCache con AWS CloudTrail .	15 settembre 2014
Nuove dimension i d'istanza supportate	ElastiCache aggiunge il supporto per ulteriori istanze General Purpose (T2). Per ulteriori informazioni, consulta Configurazione dei parametri di motore con i gruppi di parametri .	11 settembre 2014
Nuove dimension i d'istanza supportate	ElastiCache aggiunge il supporto per ulteriori istanze General Purpose (M3) e istanze Memory Optimized (R3). Per ulteriori informazioni, consulta Configurazione dei parametri di motore con i gruppi di parametri .	1 luglio 2014

Modifica	Descrizione	Data della modifica
Backup e ripristino per i cluster Redis	In questa versione, ElastiCache consente ai clienti di creare istantanee dei propri cluster Redis e creare nuovi cluster utilizzando queste istantanee. Un backup è una copia del cluster effettuata in un momento specifico che include i metadati del cluster e tutti i dati nella cache Redis. I backup vengono archiviati in Amazon S3; i clienti, quindi, possono ripristinare i dati da una snapshot in un nuovo cluster in qualsiasi momento. Per ulteriori informazioni, consulta Snapshot e ripristino .	24 Aprile 2014
Redis 2.8.6	ElastiCache supporta Redis 2.8.6, oltre a Redis 2.6.13. Redis 2.8.6 permette ai clienti di massimizzare la resilienza e la tolleranza ai guasti delle repliche di lettura, con il supporto per la risincronizzazione parziale e un numero minimo di repliche di lettura sempre disponibili, definito dall'utente. Redis 2.8.6 offre inoltre il supporto completo per publish-and-subscribe, in cui i client possono essere informati degli eventi che si verificano sul server.	13 marzo 2014

Modifica	Descrizione	Data della modifica
Motore della cache Redis	<p>ElastiCache offre il software Redis Cache Engine, oltre a Memcached. I clienti che attualmente utilizzano Redis possono «seminare» un nuovo cluster di cache ElastiCache Redis con i dati esistenti da un file di snapshot Redis, facilitando la migrazione verso un ambiente gestito. ElastiCache</p> <p>Per supportare le funzionalità di replica di Redis, l'API ora supporta i gruppi di replica. ElastiCache I clienti possono creare un gruppo di replica con un nodo di cache primario di Redis e aggiungere uno o più nodi di replica di lettura, sempre automaticamente sincronizzati con i dati della cache nel nodo primario. Nel caso di applicazioni con attività di lettura intensiva, è possibile prevedere l'offload per una replica di lettura e limitare, in tal modo, il caricamento sul nodo primario. Le repliche di lettura, inoltre, possono scongiurare la perdita di dati in concomitanza del malfunzionamento di un nodo di cache primario.</p>	3 settembre 2013
Support per Amazon Virtual Private Cloud (VPC) di default.	<p>In questa versione, ElastiCache è completamente integrato con Amazon Virtual Private Cloud (VPC). I cluster di cache dei nuovi clienti vengono creati, per impostazione predefinita, in un Amazon VPC . Per ulteriori informazioni, consulta Amazon VPC e sicurezza di ElastiCache.</p>	8 gennaio 2013

Modifica	Descrizione	Data della modifica
Support per Amazon Virtual Private Cloud (VPC)	In questa versione, ElastiCache i cluster possono essere lanciati in Amazon Virtual Private Cloud (VPC). Per impostazione predefinita, i cluster di cache dei nuovi clienti vengono creati automaticamente in un Amazon VPC; i clienti preesistenti possono passare all'Amazon VPC in qualsiasi momento, in base alle proprie esigenze. Per ulteriori informazioni, consulta Amazon VPC e sicurezza di ElastiCache .	20 dicembre 2012
Nuovi tipi di nodo di cache	Questa versione offre quattro nuovi tipi di nodi di cache.	13 Novembre 2012
Nodi di cache riservati	In questa versione è stato aggiunto il support per i nodi di cache riservati.	5 Aprile 2012
Nuova guida	Questa è la prima versione di Amazon ElastiCache User Guide.	22 agosto 2011

AWS Glossario

Per la AWS terminologia più recente, consultate il [AWS glossario](#) nella sezione Reference. Glossario AWS

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.